

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Омский государственный технический университет»

Т. В. Макарова

ВЕБ-ДИЗАЙН

Учебное пособие

*Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по направлению подготовки
09.04.02 «Информационные системы и технологии»*

Омск

Издательство ОмГТУ

2015

УДК

ББК

М15

Рецензенты:

Д. А. Хворостов, д-р. пед. наук, профессор, ОГУ

Е. А. Сидорова, д-р. техн. наук, доцент, ОмГУПС

Т.В. Макарова

М15 Веб-дизайн: учебное пособие. Омск: Изд-во ОмГТУ, 2015.– 145 с.

ISBN

Пособие ориентировано на изучение основ веб-дизайна. Рассмотрены теоретические вопросы: гипертекстовый способ представления информации в компьютерной сети; основные принципы визуального дизайна веб-страниц; разработка концепции, структуры, макета, сайта; анализ юзабилити. Изложены практические вопросы верстки веб-страниц с использованием HTML и CSS, представленные в виде комплекса учебных работ.

Учебное пособие предназначено для студентов, обучающихся по направлению «Информационные системы и технологии», профиль «Информационные технологии в медиаиндустрии». Отдельные разделы пособия могут использоваться для обучения студентов направления «Дизайн».

© Омский государственный
технический университет, 2015

Оглавление

| | |
|---|-----|
| ВВЕДЕНИЕ..... | 4 |
| ГЛАВА 1. ОБЩИЕ СВЕДЕНИЯ О ГИПЕРТЕКСТОВЫХ СИСТЕМАХ..... | 6 |
| 1.1. Гипертекст. Топология гипертекстовой системы | 6 |
| 1.2. Виды веб-сайтов | 10 |
| 1.3. Веб-сайт компании. Этапы создания веб-сайта (жизненный цикл) | 25 |
| ГЛАВА 2. ОСНОВЫ ВИЗУАЛЬНОГО ДИЗАЙНА ВЕБ-РЕСУРСОВ | 37 |
| 2.1. Основные принципы дизайна | 37 |
| 2.2. Верстка веб-страниц | 48 |
| 2.3. Адаптивный дизайн сайтов | 54 |
| 2.4. Основные аспекты улучшения качества веб-ресурса. Юзабилити..... | 61 |
| 2.5. Основные тенденции и тренды современного веб-дизайна | 69 |
| ГЛАВА 3. ПРАКТИКУМ: ОСНОВЫ HTML И CSS В ВЕБ-ДИЗАЙНЕ | 75 |
| 3.1. Основные разделы HTML-документа. Описание раздела body | 77 |
| 3.2. Форматирование текста в HTML..... | 86 |
| 3.3. Организация гиперссылок..... | 95 |
| 3.4. Добавление графики на веб-страницу. Особенности веб-графики..... | 98 |
| 3.5. Метаданные веб-документов | 107 |
| 3.6. Организация таблиц. Использование табличных структур для верстки веб-страницы..... | 113 |
| 3.7. Вложенные таблицы. Табличная верстка веб-страниц | 118 |
| 3.8. Добавление мультимедиа на страницу | 122 |
| 3.9. Стилизовое оформление веб-страницы с использованием CSS. | 128 |
| 3.10. Текстовые блоки. Верстка веб-страницы блоками <div> | 133 |
| 3.11. Проверка валидности HTML и CSS кода веб-страниц | 137 |
| РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА..... | 141 |
| ПРИЛОЖЕНИЯ..... | 142 |
| Приложение 1 | 142 |
| Приложение 2 | 143 |
| Приложение 3 | 144 |
| Приложение 4 | 145 |

ВВЕДЕНИЕ

За бурным ростом веб-технологий, обеспечивающих гипертекстовое представление и быструю передачу значительных объемов тексто-графических данных, последовало столь же стремительное развитие сферы обеспечения эстетики предъявляемой пользователям веб-сети информации и появилось новое направление профессиональной деятельности – веб-дизайн.

В понятии «веб-дизайн» отчетливо определены две его составляющие: «веб» – (*русифицированное от англ. «web» – сеть, паутина*) – ключевое слово в выражении World Wide Web, которое является названием основной службы Интернета, обеспечивающей поиск и просмотр гипертекстовых документов; и «дизайн» (*от англ. «design», – проектировать, создавать, конструировать*) – деятельность по проектированию эстетических свойств изделия с учетом его назначения и специфики использования, а также результат этой деятельности. Из сказанного следует, что веб-дизайн направлен на разработку и оформление объектов информационной среды Интернета.

В общем случае, *веб-дизайном (web page design)* называют процесс проектирования, планирования, моделирования и реализации доставки электронного содержимого через сеть (Web) с использованием технологий, подходящих для интерпретации и визуализации веб-браузером или другим графическим веб-интерфейсом пользователя.

Доставленное пользователю электронное содержимое отображается в браузере в виде веб-страницы. Именно она являет собой продукт деятельности веб-дизайнера.

Веб-страница – логическая единица Интернета. Это специальный документ, написанный на языке гипертекстовой разметки, пригодный для обработки, манипулирования и просмотра посредством веб-браузера. Контент – информационное содержание веб-страницы – включает текст, изображения, анимацию и прочие компоненты, часть которых предусматривает интерактивное взаимодействие с пользователем.

Совокупность веб-страниц, объединенных тематически, стилистически, связанных системой гиперссылок и физически находящихся на одном сервере составляют веб-сайт.

Веб-дизайнеры проектируют логическую структуру сайта и каждой его страницы, продумывают наиболее удобные решения подачи информации, проектируют художественное оформление. Эта работа требует от специалиста

комплекса профессиональных знаний и умений, включающих: владение методикой определения формальных качеств проектируемого объекта, структурирования информации, создания различных графических, стилистических, композиционных и цветовых решений, гарантирующих правильное эстетическое восприятие и удобство пользования сайтом; знание языковых средств создания гипертекстовых документов.

На веб-дизайн возлагаются несколько функций: во-первых, вид открытой страницы должен *привлечь внимание* посетителя веб-ресурса; во-вторых, дизайн должен *обеспечить удобство восприятия информации* на странице, дать ясное представление о том, из чего состоит информационный ресурс и как переместиться в требуемый раздел; в-третьих, задача веб-дизайна – *создать уникальный, индивидуальный образ*, способствующий узнаваемости веб-сайта.

В данном учебном пособии представлен материал, позволяющий получить базовые знания, необходимые для решения основных задач веб-дизайна. В первой главе даны общие сведения о гипертекстовом способе представления информации, составляющем основу популярности веб-ресурсов. Во втором представлен теоретический материал, необходимый для освоения принципов и методов работы сетевого дизайнера, осуществляющего верстку веб-страницы произвольного назначения. Третий раздел представляет собой практикум по овладению языковыми средствами веб-дизайна: языка гипертекстовой разметки HTML и языка стилевого оформления веб-страниц CSS.

ГЛАВА 1. ОБЩИЕ СВЕДЕНИЯ О ГИПЕРТЕКСТОВЫХ СИСТЕМАХ

1.1. Гипертекст. Топология гипертекстовой системы

Существует устойчивое мнение, что популярность сетевых технологий главным образом обусловлена гипертекстовым способом представления информации. Такое мнение небезосновательно, ведь ценность веб-контента (по сравнению с материалами традиционных энциклопедий) состоит в возможности мгновенного получения дополнительной информации при обращении к гипертекстовой ссылке.

Термин «гипертекст» был введен в 1965 г. Тедом Нельсоном¹ для описания документов, которые выражают нелинейную структуру идей, в противоположность линейной структуре традиционных книг, фильмов и речи. Под гипертекстом Нельсон понимал непоследовательную запись, текст, «который ветвится или осуществляется по запросу». «Обычно процесс письма осуществляется последовательно по следующим двум причинам. Во-первых, потому, что он является производным от речи..., которая не может не быть последовательной (так как у нас для этого только один канал), и, во-вторых, потому, что книги неудобно читать иначе как последовательно. Однако мысли образуют структуры, которые не являются последовательными – они связаны многими возможными переходами»².

Свои цели Нельсон описывает следующим образом: «Руководствуясь идеями, характер которых литературный, а не технический, мы создали систему для хранения и поиска текста, в котором введены взаимосвязи и «окна». Наша фундаментальная единица, документ, может иметь «окна» на любые другие документы. «Информационное тело» системы эволюционирует, непрерывно расширяясь без изменения своей основы. Новые связи и новые «окна» постоянно добавляют новые пути доступа к старому материалу».

Идея гипертекста состоит в том, чтобы дать возможность человеку воспринимать информацию в последовательности, отвечающей его интересам. В этой связи гипертекстовые документы, выстраиваются особым образом, чтобы сделать хорошо видимыми взаимосвязи-отсылки на другие документы.

¹ Тед Нельсон (Theodor Holm Nelson) -

² Landow G.P. Hypertext: The Convergence of Contemporary Critical Theory and Technology. Baltimore: The Johns Hopkins University Press, 1991.

Топология гипертекстовой сети

Возможности гипертекстовой среды заложены в основу ассоциативной навигации, предполагающей перемещение от одних информационных блоков к другим с учетом их «семантической смежности».

Разбивка информации на смысловые блоки, ее структурирование и распределение по отдельным страницам приводит к выстраиванию системы (сети), в которой отражены информационные блоки (как узлы гипертекстовой сети) и логические связи между ними.

Когда речь идет об описании компоновки сетевых компонентов, большинство профессионалов используют термин «топология», обозначающий схему соединения устройств (например, компьютеров в сети). В этом случае различают классические топологии типа: звезда, кольцо, шина. Важно понимать, что в главное топологии – это узлы и связи, а не расстояние между компьютерами и физические характеристики соединения.

Топология гипертекстовой системы – это, структура, отражающая основные магистральные связи между страницами. Существует несколько основных типов таких структур — древовидная (иерархическая), радиально-кольцевая, линейная (последовательная).

Древовидное строение гипертекстовой системы предполагает, что содержимое каждой страницы (кроме первой) входит на правах подраздела в страницу более высокого уровня. Такая структура имеет ярко выраженное начало – первую страницу, корень дерева (с него обычно начинается старт, хотя можно стартовать и из любой другой точки), но не имеет однозначного конца (страницы, посещение которой логически завершает сеанс работы с сайтом).

В такой сети можно спускаться и подниматься с уровня на уровень и перемещаться по горизонтали бесконечно долго, пока не будет решена задача или пока не исчерпается все содержимое. Древовидная структура лучше всего подходит для компонования достаточно разнородного или сложно устроенного материала — каталогов, сборников статей или ссылок (рис. 1).

Древовидное строение характерно для сайтов компаний и организаций, где отчетливо выделена главная (домашняя) страница, с которой, как правило, начинается знакомство с сайтом,

Рис. 1. Вид древовидной структуры гипертекстовой сети

Некоторым подобием древовидной структуры является *радиально-кольцевая*, примером которой может служить городская транспортная сеть (рис. 2). В

отличие от древовидной, в этой структуре больше развиты горизонтальные связи, а радиальные являются не столько иерархичными (раздел-подраздел), сколько просто отражающими направления логических переходов.

Рис. 2. Вид радиально-кольцевой структуры гипертекстовой сети

В подобной сети может присутствовать центральная точка, центр паутины, но она редко является точкой входа и, скорее, служит «пресс-центром», который обязан быть для обеспечения всевозможных формальностей. Старт, как правило, осуществляется из любой произвольной точки, движение происходит по горизонталям и вертикалям, не обращаясь к центру. Так как конца у такой системы не существует, то выходом также может становиться любая произвольная точка.

Примером такой гипертекстовой паутины является сайт проекта Википедия.

Линейные (последовательные) структуры используются, если материал, представленный в гипертекстовой системе, выстраивается в логическую цепочку — как, к примеру, главы одной книги или последовательные шаги оформления заказа в электронном магазине (рис. 3).

Рис. 3. Вид линейной (последовательной) структуры гипертекстовой сети

Линейная цепочка страниц всегда имеет логическое начало и конец, причем «запрыгивание в вагон на ходу» (т. е. вход в систему с одной из промежуточных страниц) обычно не имеет смысла, так как в этом случае тяжело поймать нить изложения. В подобном случае, пользователю приходится делать несколько шагов вверх по цепочке до достижения какого-либо понятного логического начала.

Большинство веб-сайтов имеют в своей топологии комбинацию древовидной и последовательной структуры. Например, начав с иерархического каталога товаров в интернет-магазине, после осуществления выбора, дальнейшее продвижение происходит в линейной последовательности: оформление заказа, выбор способа доставки, оплата.

Возможность генерации страниц «на лету» в ответ на запросы пользователя и применение динамического HTML для создания «живых», изменяющихся по мере прочтения и взаимодействия с ними страниц размывают стройную картину структуры гипертекстовых систем. На многих современных сайтах просто невозможно точно сказать, из скольких страниц они состоят и сколько связей содержат.

Топология веб-сайта находит отражение в визуальном оформлении его страниц и обуславливает выбор того или иного способа организации гипертекстовых переходов при проектировании системы навигации сайта.

Гипертекст может существовать только в технически сложной среде, способной обеспечить необходимую быстроту предъявления выбираемых фрагментов. В этой связи, гипертекстом часто называют не только нелинейную, сетевую форму организации текстово-графического материала, но так же и технологию, без которой невозможна такая организация в широких масштабах. Основой такой технологии является язык гипертекстовой разметки документов HTML (HyperText Markup Language).

УПРАЖНЕНИЕ 1. Определение топологии веб-сайта.

1. Выберите сайт для анализа. Исследуя навигационное меню и перемещаясь по внутренним ссылкам определите топологическую структуру сайта, изобразите ее схематически на рисунке.

Примечание: Вероятно, исследуя структуру сайта, обнаружите гибрид из двух-трех топологических схем. Выделите доминирующую.

Контрольные вопросы по теме «Гипертекст. Топология гипертекстовой системы».

1. Кто является автором термина «гипертекст»? Когда был введен термин, что означает?
2. Что называют гипертекстовой топологией сети?
3. Назовите три основные топологические структуры гипертекстовой сети. Приведите примеры сайтов, имеющих в своей основе структуру каждого типа.

1.2. Виды веб-сайтов

Веб-сайт представляет собой совокупность электронных страниц, объединенных по смыслу, связанных системой гиперссылок, и физически находящихся на одном сервере. Веб-сайт позволяет:

- создавать благоприятный имидж человека, фирмы, товара или услуги;
- обеспечивать доступность информации всем заинтересованным лицам;
- реализовывать возможности мультимедийного представления информации (с использованием графики, звука, анимации и видео);
- оперативно реагировать на различные ситуации: обновлять данные, анонсировать события, обеспечивать информационную поддержку проводимых мероприятий;
- предлагать услуги и продавать продукцию через Интернет;
- организовывать обратную связь и общение с людьми.

Классификация веб-сайтов

Веб-разработчики создают оригинальные и уникальные интернет-сайты различной направленности и различного уровня сложности. При этом многие сайты имеют схожие цели, задачи, способы представления информации и т.п.

На основании схожих целей, контента и типа представления данных, сайты разделяют на группы. Для этого выделяют некоторые характерные принципы организации сайтов, которые выступают в качестве основных классификационных критериев, внутри которых происходит деление.

Веб-сайты можно классифицировать по следующим признакам:

По возможностям взаимодействия с контентом.

В этой категории сайты разделяют на:

- информационные (документно-ориентированные) сайты

Подобные сайты предоставляют информацию и обеспечивают ограниченный уровень интерактивности (есть возможность просмотра, поиска и сортировки предоставленной информации).

- проблемно-ориентированные сайты

Сайты позволяют взаимодействовать с информацией для выполнения какой-либо задачи (например, купить авиабилет или пообщаться с коллегой). Страницы таких сайтов могут генерироваться в соответствии с запросом

пользователя. Например, так формируется страница результатов поиска, информация для которой извлекается из специальной базы данных (рис. 4).

- гибридные сайты

Объединяют возможности двух первых типов.

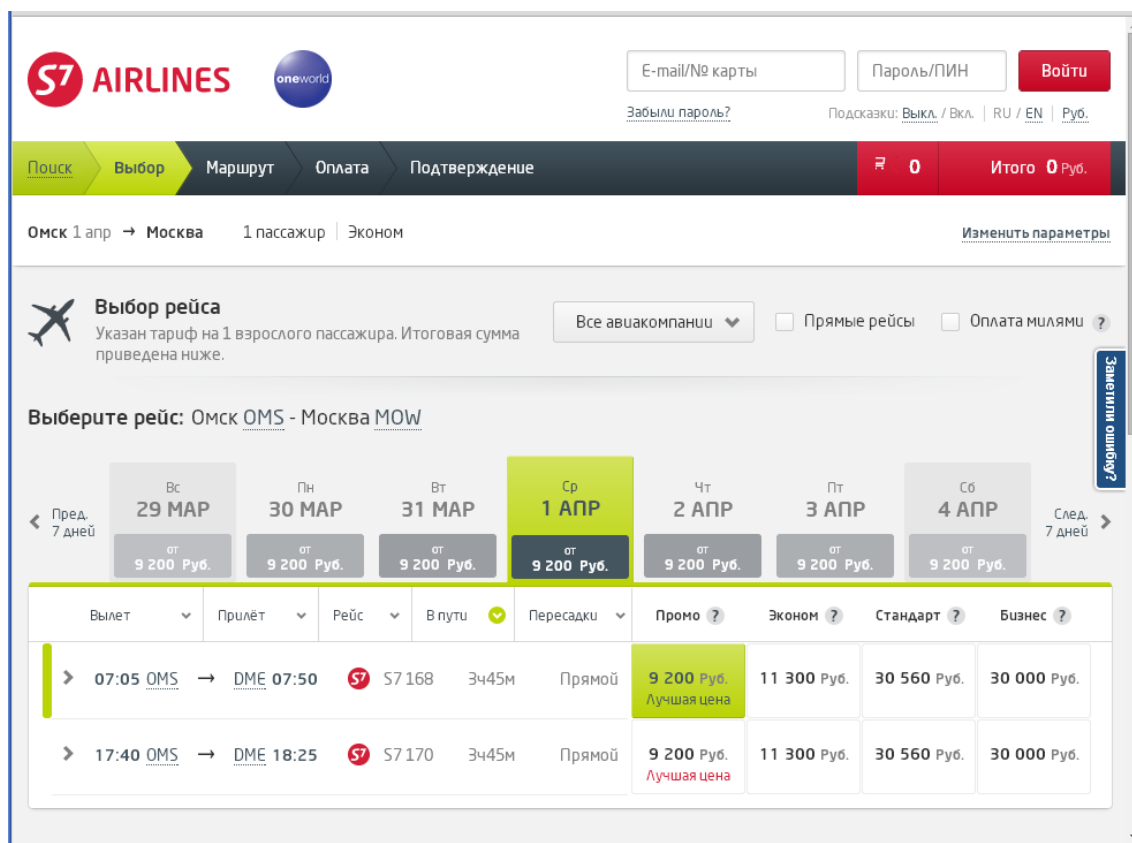


Рис. 4. Страница проблемно-ориентированного сайта авиакомпании (<http://www.s7.ru>) сформирована в ответ на запрос о поиске рейсов на указанные даты вылета

По частоте изменений контента.

В зависимости от частоты обновления информации, различают:

- статические сайты

К статичным относят сайты, страницы которых создаются однажды и в течение длительного времени не изменяются.

- динамические сайты

Динамичными называют сайты, содержимое которых со временем изменяется (обновляется). Содержимое может модифицироваться на регулярной основе (с

заявленной периодичностью, например, ежедневно, еженедельно, ежемесячно) или по мере поступления свежей информации (рис. 5).

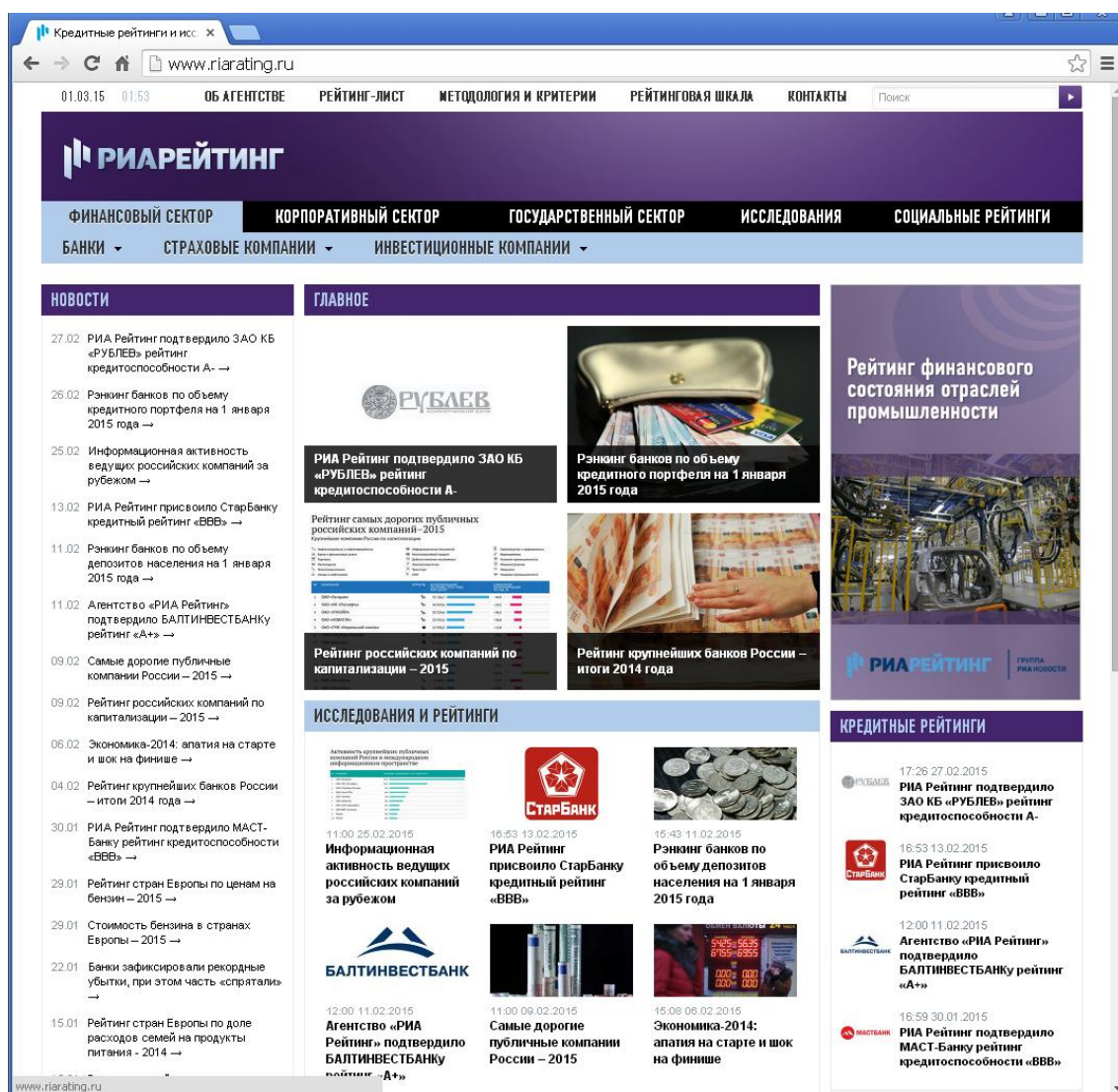


Рис. 5. Сайт новостного агентства РИА-новости (<http://www.ria.ru/>) имеет динамическую структуру

По назначению.

По основному предназначению, сайты можно разделить на:

- коммерческие сайты

Коммерческие сайты создаются преимущественно для поддержания бизнеса какой-либо организации. Основной аудиторией сайта являются потенциальные и настоящие клиенты организации, инвесторы, партнеры. Среди основных целей существования коммерческих сайтов можно выделить: распространение основной информации о компании, ее продукции и услугах, установление связи с инвесторами и общественностью, поиск служащих, электронная коммерция и др. (рис. 6).

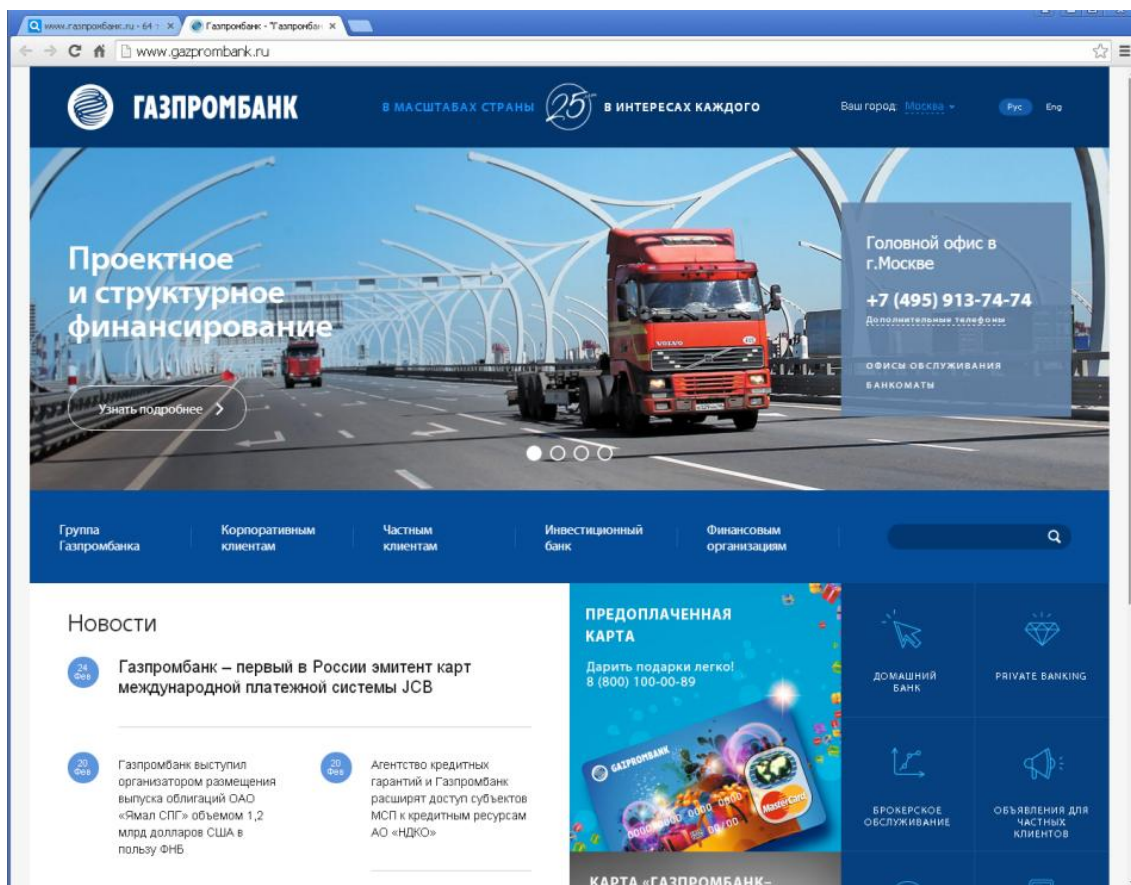


Рис. 6. Главная страница коммерческого сайта АО Газпромбанк (<http://www.gazprombank.ru/>)

– информационные сайты

К информационным сайтам относятся, как правило, правительственные, образовательные, новостные сайты, а также сайты некоммерческих организаций, религиозных групп и т.д. В основе управления могут лежать некоторые коммерческие факторы, но основная цель, заключается в предоставлении информации. Аудитория сайта состоит из тех, кто имеет интерес или необходимость обратиться за информацией к авторитетному Интернет-источнику (рис. 7).

– развлекательные сайты

Развлекательные сайты могут быть коммерческими и некоммерческими. Главное их предназначение – развлечь своих посетителей, поэтому продаваемым товаром является само развлечение (рис. 8).

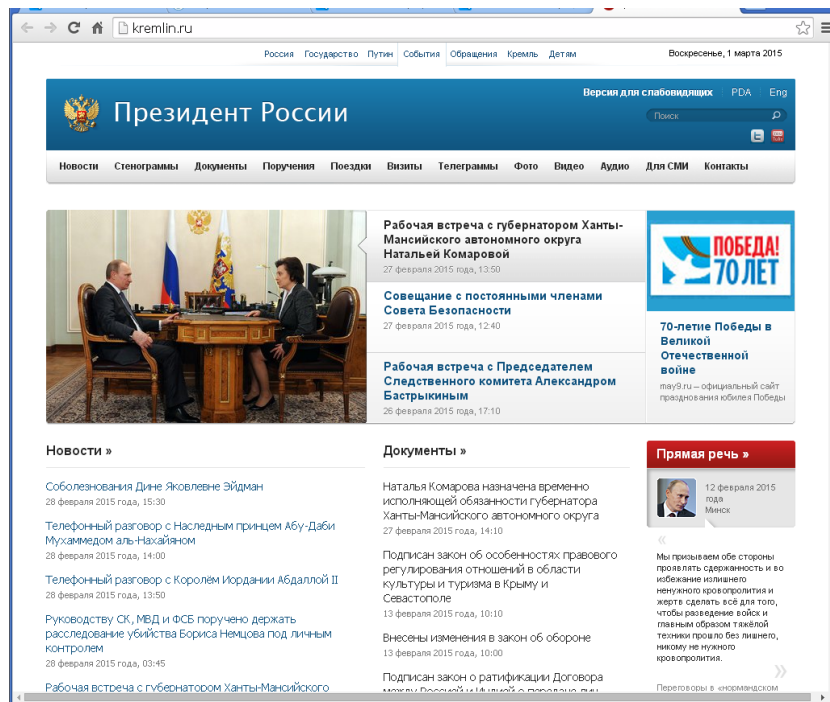


Рис. 7. Информационный сайт Президента Российской Федерации (<http://kremlin.ru/>)

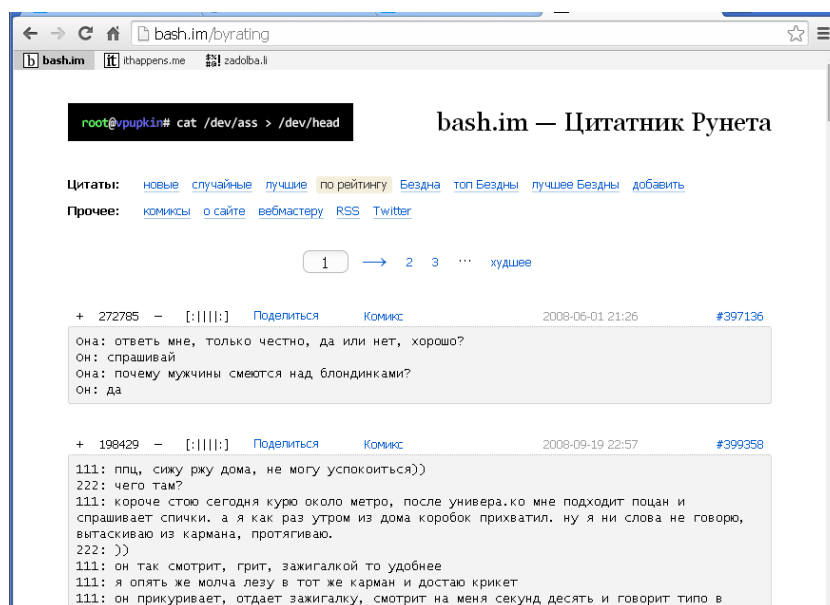


Рис. 8. Страница развлекательного сайта – «Цитатник Рунета» (<http://bash.im/>)

– общественные сайты

Общественные сайты предоставляют возможность простого взаимодействия между посетителями и служат для них местом встреч, (гостевые книги, форумы и т.д.). На основе общественных сайтов строятся так называемые социальные сети. Интерактивные страницы, подобные площадкам общественных сайтов, могут присутствовать также на сайтах других видов, при этом, наличие

форумов, чатов или гостевых книг не означает, что сайта относится к категории общественных.

– навигационные сайты

Навигационные сайты призваны оказывать пользователям помощь в поиске нужной информации в сети Интернет. К этому типу сайтов относят порталы, являющиеся своеобразными центрами, размещающие указатели на другие точки в Интернете, а также поисковые системы и каталоги сайтов (рис. 9).

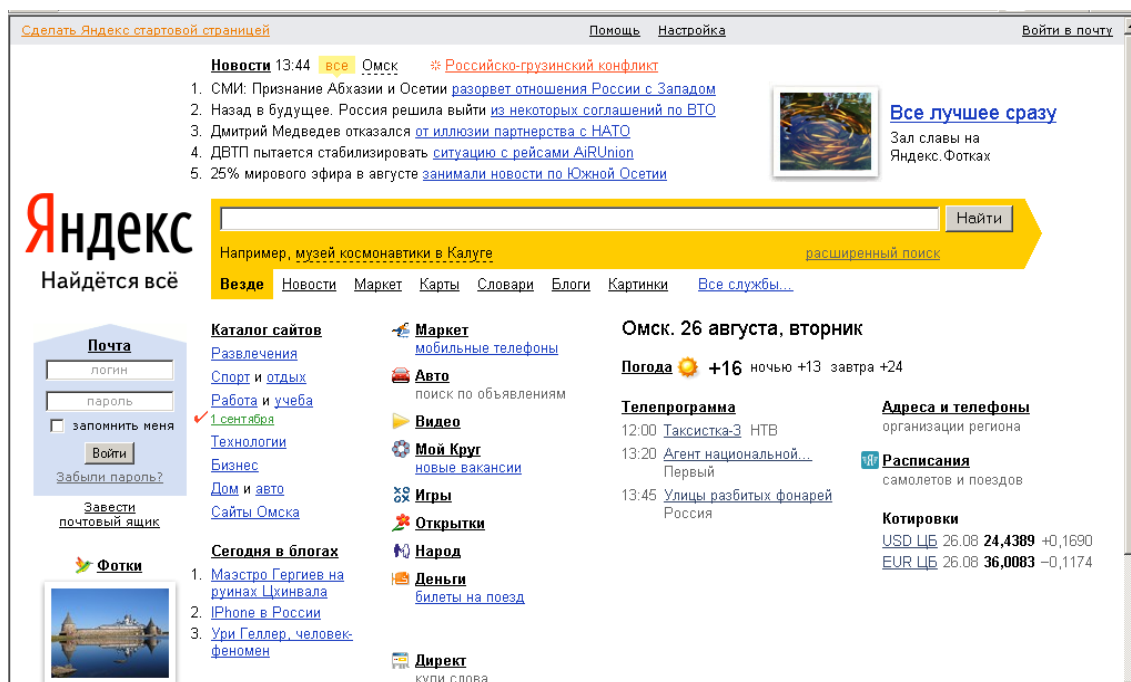


Рис. 9. Главная страница поисковой системы «Yandex.ru» (<http://www.yandex.ru>)

– художественные сайты (электронные выставочные площадки)

Художественные сайты являются предметом самовыражения художника и служат для того, чтобы вдохновить, просветить или развлечь его посетителей (рис. 10).

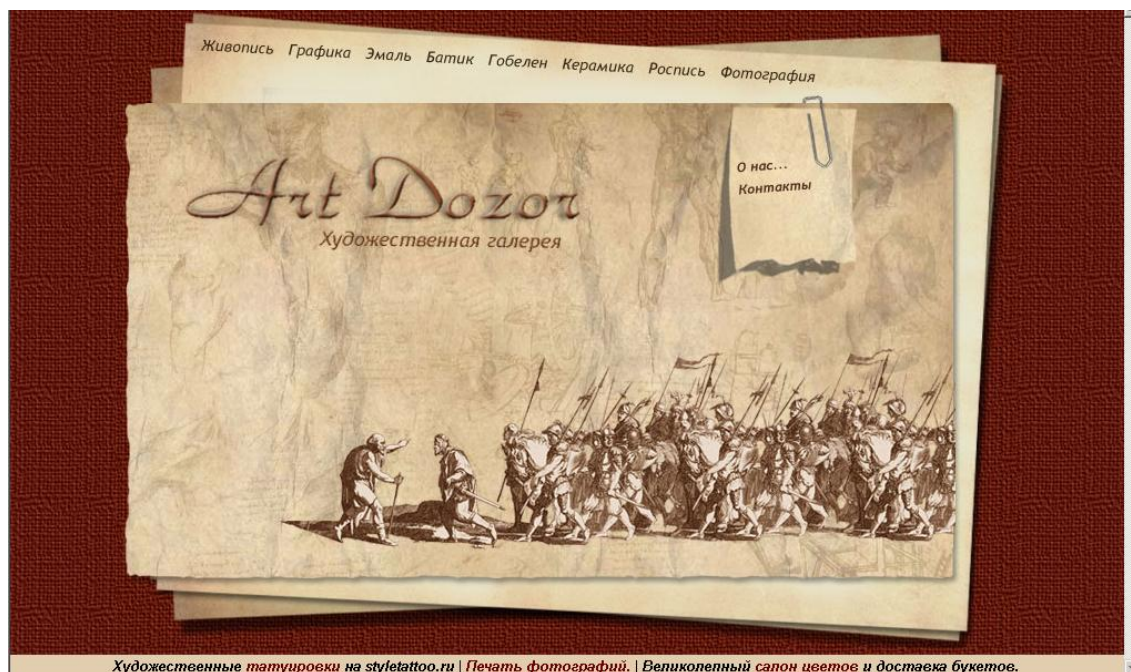


Рис. 10. Художественный сайт галереи «Art Dozor» (<http://artdozor.ru>)

Галерея современной живописи «Art Bay» (<http://www.art-bay.ru/>)

По характеру решаемых маркетинговых задач.

Коммерческие сайты по характеру решаемых маркетинговых задач и, соответственно, уровню интерактивности, сайты можно разделить на следующие типы:

- сайт-визитка (сайт-буклет, сайт-каталог)

К данной категории относятся простые сайты, состоящие из нескольких статичных страниц, имеющих общий дизайн-шаблон, основанный на фирменном стиле компании, на которых представлена минимально-необходимая (чаще всего только вводная и контактная) информация о фирме и продукции (рис. 11). Сайт-визитка подходит для небольших фирм или творческих групп, а также используется тогда, когда основной сайт находится на реконструкции или в процессе создания, либо когда руководство решает, что контактной информации о компании в сети Интернет достаточно для первого знакомства с клиентом.

Сайт-буклет и сайт-каталог отличаются от сайта-визитки большим количеством информации (более подробной) о товарах и услугах.



Рис. 11. Главная страница сайта-буклета ресторана «У Швейка» (<http://www.shveikomsk.ru/>)

– Интернет-представительство

Интернет-представительствами принято называть информационные, (как правило, коммерческие) сайты, содержащие полную информацию об организации: цели, задачи, политика, принципы работы, планы на будущее и т.д. Такие сайты создаются для решения задач формирования имиджа, привлечения новых клиентов, завоевания новых рынков и т.п. На страницах Интернет-представительства могут быть размещены каталоги продукции, подробная информация о товарах и услугах, новости, гостевые, форумы, система «вопрос-ответ» и т.п. Данный тип сайта подходит средним и большим компаниям, желающим расширить сферы своей деятельности. Интернет-представительство обычно имеет простой, но элегантный дизайн на основе фирменного стиля компании (рис. 12).

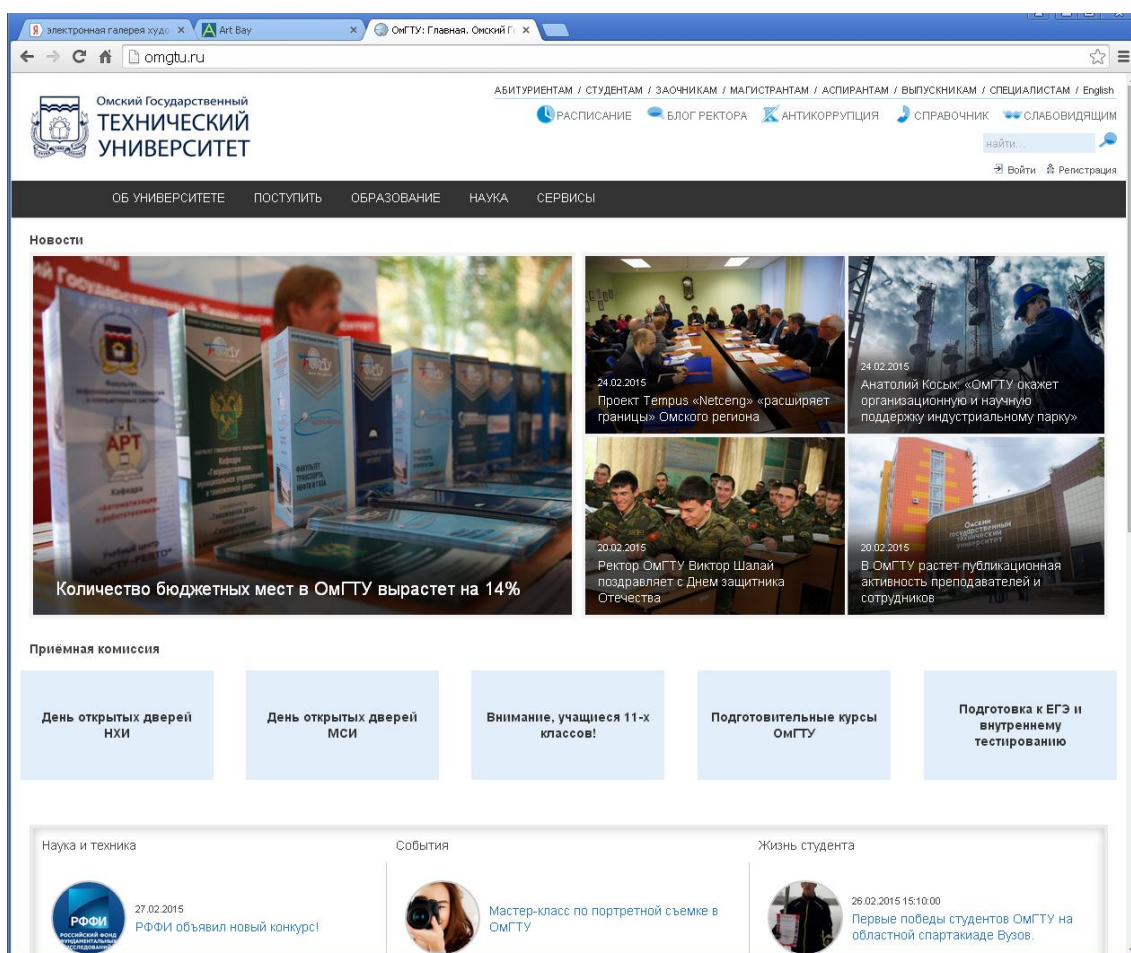


Рис. 12. Главная страница сайта ОмГТУ, выполняющего функции Интернет-представительства (<http://www.omgtu.ru/>)

– промо-сайт

Промосайтом называют небольшие по размеру сайты, предназначенные для информационного освещения проводимой акции, продвижения какого-либо товара или услуги. Такие сайты создаются, например, если компания продвигает на региональный рынок новый товар, в этом случае, промо-сайт будет посвящен только ему. Такие сайты обычно имеют яркий креативный дизайн с графикой, анимацией, оригинальной системой навигации и т.п. На таком сайте нет новостей или сложных программных модулей, но зато он привлекателен и хорошо запоминаем (рис. 13).

Промо-сайты чаще всего создаются только на период проведения мероприятия.

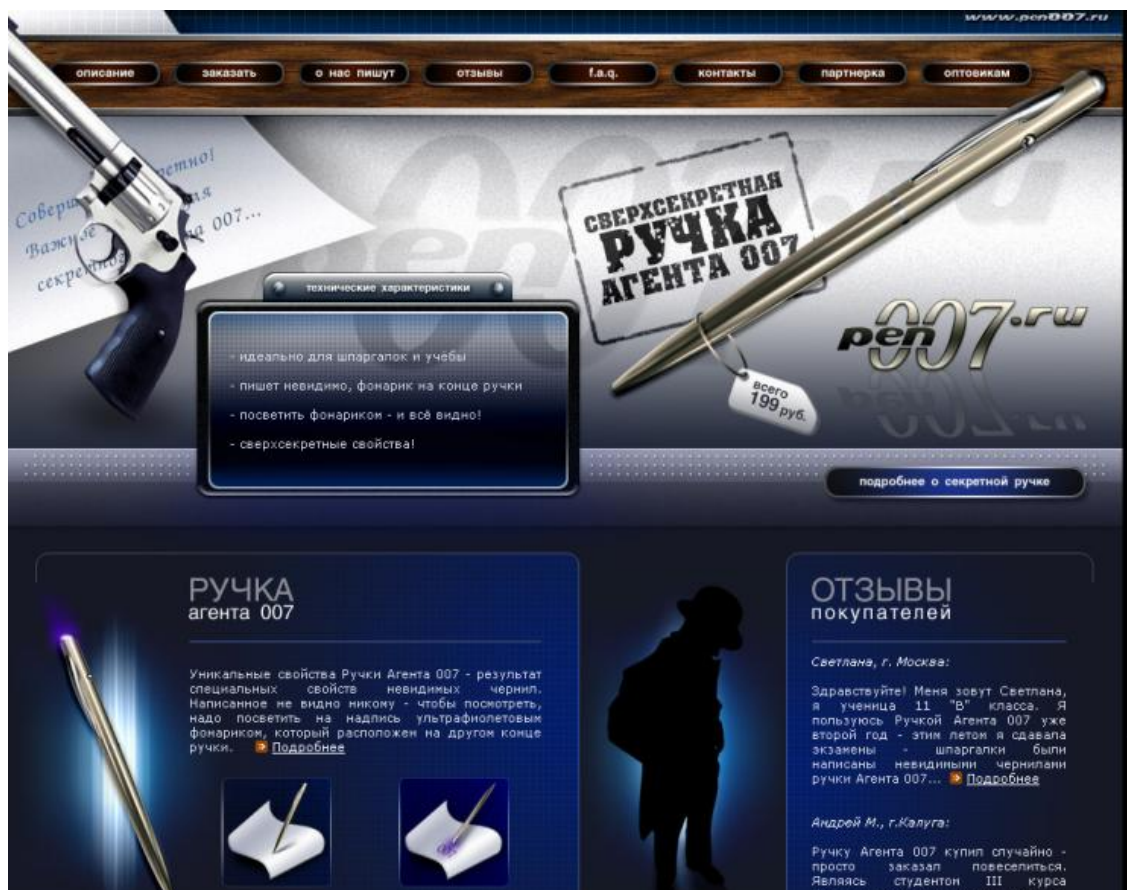


Рис. 13. Промо-сайт, посвященный продвижению нового товара – «Ручка агента 007» (<http://pen007.ru>)

– интернет-магазин

Интернет-магазин или «электронный супермаркет» – это самостоятельный сайт или группа страниц, которая может быть реализована вместе с любым другим видом сайта, начиная от визитки и заканчивая порталом. Интернет-магазин содержит товарные группы, позиции и цены товаров, страницы с потребительскими корзинами, формы заказов и др.

– порталы

Порталами называют большие, со сложной внутренней структурой ресурсы, посвященные какой-либо теме, товару и т.д. (например, сотовой связи, поэзии, медицине и т.п.). Портал, как правило, содержит большое количество информации по определенной тематике, включая каталоги, списки телефонов, обзоры, статьи, форумы, ссылки на аналогичные по тематике сайты и т.п. (рис. 14).

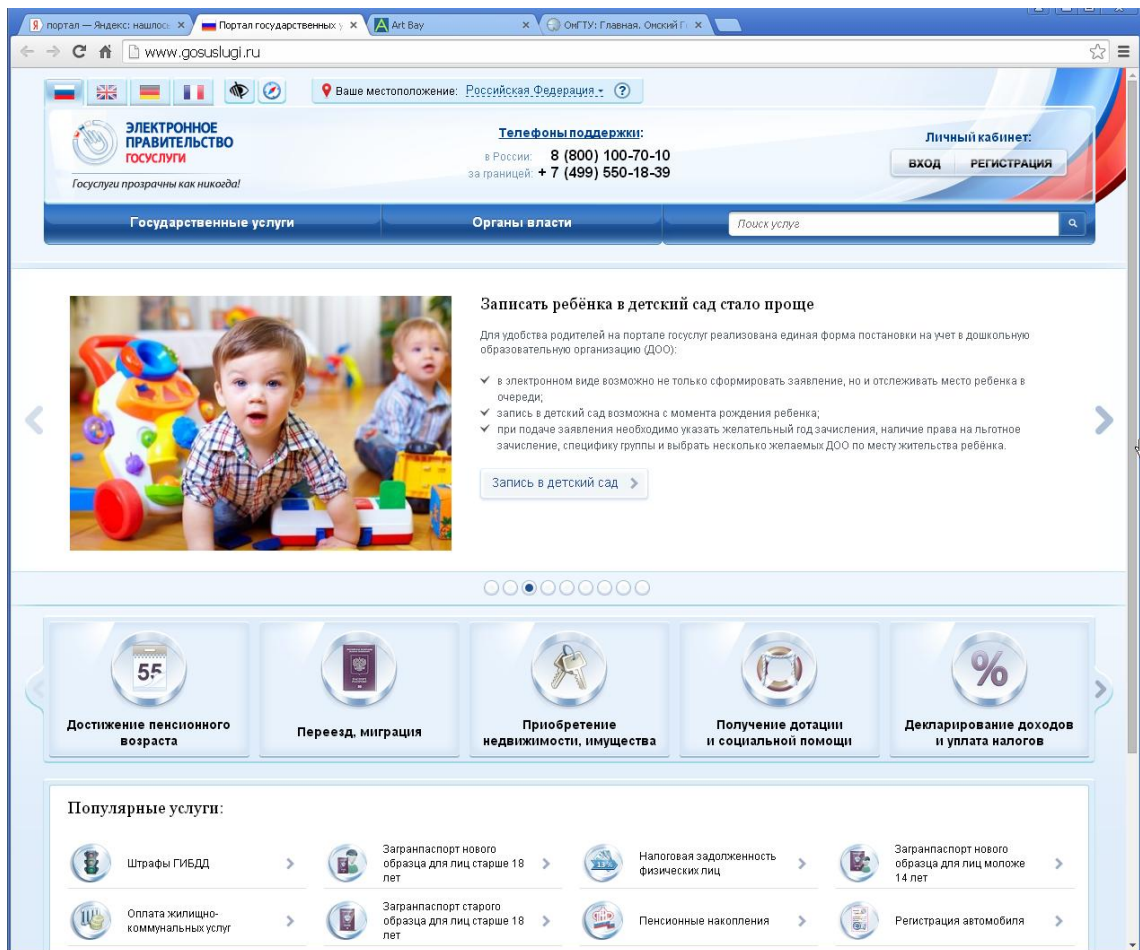


Рис. 14. Главная страница портала государственных услуг (<http://www.gosuslugi.ru/>)

По типу дизайна (исполнения) сайта

С точки зрения визуального исполнения можно назвать следующие виды сайтов:

- текстово-ориентированные сайты

Текстовый дизайн сайта используют в случае, когда необходимо сосредоточить внимание на текстовом содержимом. Для подобного типа дизайна характерным является минимальное использование графики, а расстановка смысловых акцентов достигается применением выделений шрифтом и цветом (рис. 15).

Текстовый дизайн рекомендуется применять для тех сайтов, где важны скорость загрузки и гибкость в отображении информации на странице.



Рис. 15. Для сайта, посвященного рекламе выбран вариант текстового дизайна (<http://www.advertology.ru>)

– стилизованные под пользовательский интерфейс

Один из самых популярных способов организации сайтов можно назвать «сайты, действующие как обычные программные приложения». Дизайн таких сайтов стилизован под GUI (graphical user interface – графический пользовательский интерфейс), то есть напоминает оформление пользовательского интерфейса, большинства программных продуктов среды Windows. Навигационные элементы представлены как меню, состоящее из разделов с вложенными подразделами и страницами, что повторяет способ организации документов большинства программных приложений (рис. 16).

Этот тип дизайна предпочитают опытные пользователи, поскольку он является привычным, удобным и интуитивно понятным, что способствует увеличению скорости работы с сайтом и облегчению поиска нужной информации.



Рис. 16. Навигация сайта организована по принципу традиционного интерфейса большинства пользовательских приложений (<http://www.zve.ru>)

– метафорические сайты

Оформление такого рода сайтов основано на метафоре, то есть на использовании визуальных образов, связанных с образами реального мира. При метафорическом сравнении с объектами реального мира, знакомыми и понятными для посетителей, общение с сайтом становится простым и интуитивно понятным для не только для опытных, но и для начинающих пользователей. Однако, здесь существует опасность — не все пользователи смогут понять метафору.

Метафоричность проявляется главным образом в организации системы навигации по сайту. Такое исполнение привлекает внимание и хорошо запоминается, поэтому подходит для художественных или промо-сайтов, а также редкопосещаемых страниц (рис. 17). Сайты, созданные с привлечением метафоры часто выполняют рекламные функции благодаря яркому запоминающемуся дизайну и навигации.



Рис. 17. Дизайн страниц сайта основан на метафоре, которой является книга, с вложенными закладками для удобного поиска и открытия наиболее нужных страниц (<http://nipr.tel86.ru>)

– нетрадиционный дизайн

Сайты с экспериментальным или нетрадиционным дизайном отличаются несоблюдением каких-либо соглашений, нарушением правил, предоставлением содержимого и навигации в неожиданном варианте. Подобный тип дизайна используется чаще всего на сайтах, посвященных развлечениям, искусству, рекламе и т.п.

УПРАЖНЕНИЕ 2. Определение топологии веб-сайта.

1. Выберите сайт для анализа. Классифицируйте сайт по: возможностям взаимодействия с контентом, частоте изменений контента, назначению, характеру решаемых маркетинговых задач (для коммерческих сайтов), типу дизайна (исполнения) сайта.
2. На основе анализа сделайте вывод о соответствии контента и оформления сайта целевой аудитории посетителей этого ресурса и решаемым посредством сайта задачам.

УПРАЖНЕНИЕ 3. Выявление общих тенденций в оформлении сайтов одной категории.

1. Отыскать в Интернете 5-10 примеров сайтов одной из категорий: сайт-визитка, интернет-представительство, промо-сайт, портал.
2. Выявить общие тенденции в оформлении сайтов выбранной категории.

Контрольные вопросы по теме «Виды веб-сайтов».

1. Что представляет собой веб-сайт?
2. По каким признакам классифицируют Интернет-сайты?
3. Какие задачи выполняет веб-сайт компании?

1.3. Веб-сайт компании. Этапы создания веб-сайта (жизненный цикл)

Планируя разработку и запуск веб-проекта, следует начинать с главного – определения его цели. Определение целей создания сайта необходимо для лучшего понимания истинных задач, которые владелец собирается решать с помощью сайта. Правильно и четко сформулированная генеральная цель позволит поставить конкретные задачи, решение которых приведёт к превращению сайта в эффективный инструмент рекламной, коммерческой или иной деятельности.

Задачи сайта

Компании могут преследовать различные цели присутствия сайта в сети Интернет, например:

1. Создание инструмента распространения информации о компании и ее бизнесе. Полноценный Интернет-сайт компании предоставляет документы, фото, аудио- и видеоматериалы, необходимые для формирования правильного представления о возможностях и деятельности компании, всем желающим.

2. Помощь компании в поддержании старых и установлении новых связей на рынке. Информационная среда Интернета упрощает процесс деловой переписки и общения. Кроме того, создание сайта позволяет избежать стандартных процедур ознакомления с деятельностью компании на ее территории.

3. Информационная и сервисная поддержка клиентов и партнеров. Посредством Интернет-сайта можно не только оперативно отвечать на вопросы, возникающие у пользователей продуктов и услуг компании, но и формировать базу часто задаваемых вопросов. Тем самым удастся избежать многократных разъяснений по решению одних и тех же задач.

4. Увеличение объема продаж и поиск новых покупателей продуктов компании. Компания, успешно работающая на традиционном рынке, при помощи Интернет-сайта может охватить большую целевую аудиторию, хотя бы потому, что расстояние не играет в сети Интернет такого принципиального значения, как в реальной жизни.

5. Реализация удаленной демонстрации товара для продвижения продаж (sales-promotion). На Интернет-сайте компании может быть представлена развернутая информация о товарах и услугах. Помимо общего описания в рамках ресурса возможна публикация рейтингов и аналитических отчетов, организация консультаций клиентов, партнеров и дилеров.

6. Осуществление информационной поддержки и информационного обмена с удаленными представительствами и торговыми точками. Для компании с региональными представительствами или несколькими торговыми точками в одном городе, Интернет может стать альтернативным каналом обмена информацией, обладающим более высокой скоростью, чем традиционные. При этом Интернет позволяет одинаково успешно передавать различную информацию: документацию, презентационные материалы, инструкции, внутренние документы и т.д.

7. Обеспечение круглосуточного предоставления услуг клиентам и партнерам. Интернет-сайт функционирует не разделяя время на «рабочее» и «нерабочее», поэтому к нему можно обращаться круглосуточно. Таким образом, в любое урочное и неурочное время, запросы клиентов, которым потребуются услуги или информация о бизнесе компании будут удовлетворены.

8. Организация оперативной обратной связи с клиентами в любой точке мира. Создание сайта позволяет компании не только предоставлять информацию посетителям, но и проводить маркетинговые исследования. Анализ результатов опросов, интерактивного общения, а также статистики посещений помогут компании определить эффективность отдельных мероприятий, скорректировать работу сайта и максимально приблизить информационное наполнение ресурса к запросам целевых аудиторий.

9. Взаимодействие с традиционными и электронными средствами массовой информации. СМИ проявляют повышенный интерес к Интернету как источнику первичной информации, поэтому сайт может использоваться как канал воздействия на журналистов.

Сайт, представляющий в Интернете компанию, должен не просто нести информацию и делать рекламное или коммерческое предложение от лица организации, но, наряду с этим, быть ценным для пользователя. Он должен не только обозначать факт присутствия в сети, а формировать новый позитивный взгляд на продвигаемую марку/услугу/прочее.

Жизненный цикл информационной системы

Веб-сайт представляет собой информационную систему (ИС), жизненный цикл которой включает в себя все этапы развития от возникновения потребности и определенного целевого назначения сайта, до полного прекращения его использования вследствие морального старения или потери необходимости решения соответствующих задач.

Понятие жизненного цикла является одним из базовых в методологии проектирования информационных систем. Жизненный цикл системы – процесс создания и сопровождения сайта – представляется перечнем этапов (стадий) и выполняемых на них процессов. Для каждого этапа определяются временные рамки, состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т. д. Такое формальное описание жизненного цикла ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Полный жизненный цикл информационной системы, как правило, включает в себя:

- стратегическое планирование,
- анализ,
- проектирование,
- реализацию,
- внедрение,
- эксплуатацию,
- утилизацию.

Структура, определяющая последовательность выполнения этапов работ и взаимосвязи процессов, действий и задач на протяжении жизни ИС называется *моделью жизненного цикла информационной системы*.

Модель ЖЦ ИС включает в себя:

- стадии (часть процесса создания ИС, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (техническое задание, макет веб-страниц, пр.), определяемого заданными для данной стадии требованиями);
- результаты выполнения работ на каждой стадии;
- ключевые события (точки завершения работ и принятия решений).

В настоящее время в проектировании информационных продуктов используются следующие модели жизненного цикла.

Каскадная модель (рис. 18) предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

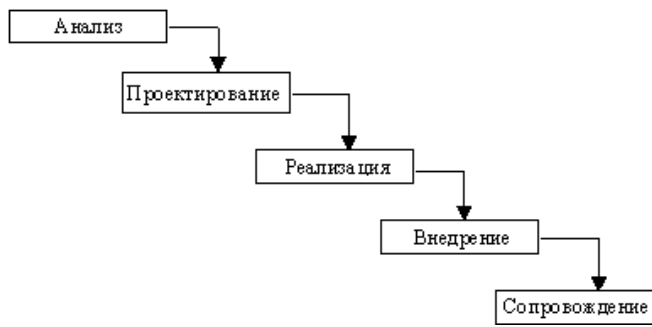


Рис. 18. Каскадная модель жизненного цикла ИС

Поэтапная модель с промежуточным контролем (рис. 19). Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

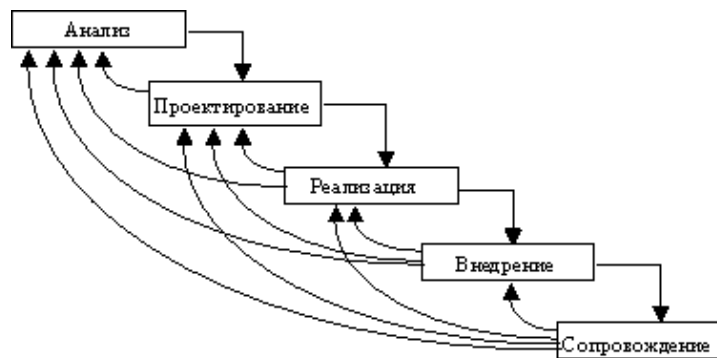
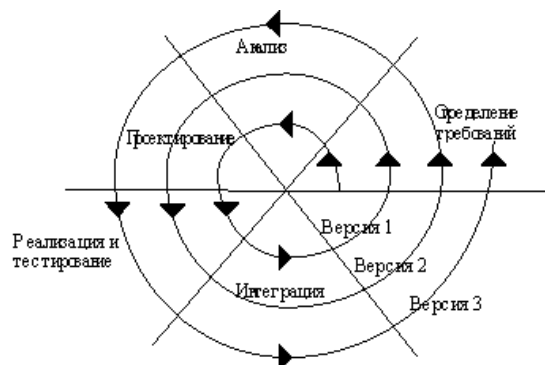


Рис. 19. Поэтапная модель с промежуточным контролем

Спиральная модель (рис. 20). На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество, и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки - анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).



В практике создания веб-ресурсов наибольшее распространение получили две основные модели жизненного цикла: каскадная и спиральная модели.

Аудитория веб-сайта

При создании сайта важно понимать для кого он создаётся, то есть обозначить характеристики целевой аудитории. При описании аудитории учитываются: география проживания; демографические характеристики (национальность, язык, пол, возраст, образование, род занятий и т. п.); уровень технической подготовки; конфигурация компьютера, тип соединения с сетью, тип используемого браузера; пути попадания пользователей на сайт; задачи, которые пользователи пытаются решить посредством сайта; время и продолжительность посещения сайта; число и причины повторных визитов, предпочтения, тип поведения пользователей др. Помимо основных, перечисленных выше, характеристик аудитории, важно также определить причины, побуждающие пользователей обращаться к сайту и то, каким образом посетители решают свои задачи.

Знание целевой аудитории позволяет сформулировать конкретные и измеримые цели существования сайта, а также его дизайн-концепцию.

Составляющие процесса создания рекламно-информационного сайта

Процесс создания веб-сайта включает в себя несколько этапов:

- выбор доменного имени сайта;
- определение информационного наполнения (контента);
- разработку структуры и системы навигации по сайту;
- разработку дизайна сайта;
- разработку инструментов работы с аудиторией (интерактивности).

Выбор доменного имени (адреса) и названия сайта

Чаще всего, посетители попадают на сайт, просто угадывая его название, исходя из своих представлений о существовании некой услуги или торговой марки. Кроме того, на сайт могут попасть те пользователи, кто-то уже посещал ресурс (например, перейдя на него по ссылке) и запомнил его название с целью повторного посещения. Помимо этого, аудитория сайта активно пополняется из числа тех, кто, обратил внимание на Интернет-адрес компании, указанный в рекламных материалах (на радио, в буклетах, в газетах, визитках и т.п.) в оф-лайн среде.

В описанных примерах, да и в реальной практике, посещаемость сайта напрямую зависит от правильности выбора доменного имени³. Грамотный подход к выбору доменного имени не только способствует привлечению дополнительных пользователей, но и становится одной из наиболее значимых частей маркетинговой кампании.

Для выбора доменного имени существуют следующие рекомендации:

- доменное имя должно однозначно ассоциироваться с компанией – владельцем сайта (содержать название компании либо указывать на сферу её деятельности и т.п.);
- имя должно быть достаточно коротким и легко произносимым;
- доменное имя должно быть простым в написании (во избежание ошибок пользователей при его наборе) и легко запоминаться.
- доменное имя не должно быть похожим на имя конкурентов (в случае его неправильного написания).

Соблюдение данных рекомендаций помогает получить дополнительный приток посетителей за счет более легкого обнаружения веб-сайта пользователями, обладающими первоначально малым объемом информации об искомой организации.

По адресам <http://www.register.com> (для доменных зон: .com, .org, .net, .org) и <http://www.ripn.net/nic/whois/> (для доменной зоны .ru) можно проверить, зарегистрировано ли уже на кого-либо выбранное имя. Кроме того, проверить, занято ли выбранное доменное имя, а также получить список только что освободившихся доменов, можно на сайте www.webnames.ru.

Определение информационного наполнения сайта (контента)

Одной из самых важных составляющих сайта является контент.

Контент – это информационное наполнение сайта: тексты, иллюстрации, интерактивные элементы, подобранные и упорядоченные в соответствии с заранее продуманной концепцией и структурой сайта. Собственно, контент – это то, ради чего посетители приходят на сайт.

Наполнение сайта тематической информацией, выполняет одновременно две важные функции: повышение ценности и значимости сайта по сравнению с конкурентами; повышение его значимости для поисковых систем.

³ Дополнительная информация в книге: Серго А. Г. Доменные имена: монография. – М.: Российская государственная академия интеллектуальной собственности (ФГБОУ ВПО РГАИС), 2013. – 312 с.

Различают «стартовое» информационное наполнение веб-сайта, которое проектируется на этапе создания, и текущее информационное наполнение, которое организуется на этапе эксплуатации веб-сайта в рамках работ по сопровождению (поддержке) или модернизации сайта.

В работе по созданию контента учитываются следующие факторы:

- тип, формат представления и полнота информационного наполнения интернет-ресурса;
- удобство восприятия и поиска информации на страницах сайта (структурирование), корректность отображения информации на различных программных платформах (на компьютерах посетителей сайта);
- стиль изложения информации, соответствие стиля особенностям основной целевой аудитории;
- акценты в представлении информации, определяемые логикой подачи материалов;
- предполагаемая периодичность обновления и добавления информации;
- использование ключевых фраз, поисковых слов и т.п., требуемых для продвижения ресурса.

При разработке контента особые требования предъявляются к форме подачи информации. Вызвано это рядом объективных причин:

особенностью поиска информации в сети

В Интернете, в отличие от прочих масс-медиа, продукция конкурентов (сайты сходной тематики) присутствует в огромном количестве и находится на расстоянии одного щелчка мыши. Поэтому важно максимально быстро завладеть вниманием пользователя и постараться удержать его.

Из-за дефицита внимания, в Интернете часто практикуется пирамидальная структура представления информации: сначала главное, ниже - подробности. Самый важный материал, дающий представление о теме, выносится в начало страницы (на первый экран). Кроме того, применяется представление информации посредством заголовка и краткого анонса - если интересно, нажми на ссылку «подробнее» и прочитай полный текст.

особенностью восприятия информации на экране

Человеку привычно считывать информацию не с экрана, а с печатного листа, когда глаз скользит по коротким текстовым строкам, а движение глаза направляют специальные засечки на символах типографских шрифтов. Текст

же экранной строки имеет большее количество символов (длину строки) и, как правило, оформляется шрифтом без засечек, поэтому чтение с экрана быстро утомляет и большинство пользователей избегают читать с экрана длинные тексты.

Для удобства чтения, большие текстовые блоки на экране уместно разбивать на небольшие абзацы, разделённые между собой пустыми строками, и дублировать «версией для печати», которую, напротив, рекомендуется делать «нормального» размера, а не дробить на небольшие куски. Кроме того, в целях облегчения восприятия, рекомендуется расставлять смысловые акценты, используя выделение цветом, размером шрифта и т.п.

необходимостью оплачивать время, проведённое в сети, либо полученный объем информации.

Несмотря на большое разнообразие способов подключения к Интернету и вариантов оплаты, можно утверждать, что только пользователи с подключением, предполагающим безлимитный трафик и достаточно высокую скорость передачи данных, не испытывают тревогу относительно времени загрузки и размера (в байтах) страницы сайта, а так же быстроты поиска на нем нужной информации.

Чтобы знакомство с сайтом было наиболее комфортным для пользователя, следует делать так, чтобы уже по главной странице пользователь смог понять, что представляет собой сайт и чем будет интересен и полезен. В связи с этим, на главной странице уместно помещать анонсы новостей, оповещать о новых продуктах и маркетинговых программах.

Очень важным является и размер файлов загружаемых документов (страниц). Нельзя допустить потери посетителя только потому, что он не смог дождаться окончания загрузки сайта. Желательно не делать объем заглавной страницы более 70 килобайт (у прочих страниц размер должен быть еще меньше). Графика, которая используется, на сайте, должна быть максимально оптимизирована в соответствии с данным требованием.

В настоящее время особое внимание уделяется возможности персонализации контента. С помощью средств персонализации пользователь может сам формировать состав и форму подачи ему информации на том либо ином сайте, включать только интересные ему новости, получать информацию в удобном для него оформлении и т.д.

Разработка системы навигации по сайту

Разбивка информации на смысловые блоки, ее структурирование и распределение по отдельным страницам приводит к необходимости создания продуманной системы навигации веб-сайта.

Система навигации веб-сайта представляет собой совокупность гипертекстовых ссылок, связывающих воедино все страницы сайта по определенному принципу. Главное ее предназначение состоит в том, чтобы сориентировать пользователя при поиске информации на сайте и оказать помощь в перемещении по сайту.

Основные требования к системе навигации:

- навигация должна быть интуитивно понятной для пользователя с любым уровнем подготовки;
- должна обеспечивать быстрый поиск необходимой посетителю информации;
- должна давать представление о том, из чего состоит сайт и что полезного можно в нем найти.

Существует целый набор разноплановых средств навигации, каждое из которых удобно для определенного круга пользователей, либо для конкретного типа поиска информации на сайте. При этом, некоторые элементы навигации дублируют друг друга.

Основными группами навигационных элементов являются: навигация сверху (снизу), навигация слева, навигация справа.

Навигация сверху в виде меню удобна и популярна, поскольку напоминает интерфейс большинства пользовательских программных приложений. В ней содержится перечень разделов и подразделов сайта, который, как правило, дублируется также снизу страницы, чтобы избавить пользователя, достигшего конца страницы и желающего продолжить изучение сайта, от лишней прокрутки.

Навигация слева страницы является самой распространенной для людей читающих слева направо, поскольку находится в начале пути следования глаз читающего.

Навигация справа используется также довольно часто, поскольку расположение рядом с полосой прокрутки удобно для пользователей, нацеленных на быстрое перемещение по сайту, т.к. позволяет сократить расстояние, которое пробегает мышь.

Помимо перечисленных основных групп, навигационные функции выполняют такие элементы как:

- *пиктограммы* (наглядно, посредством визуального образа, демонстрируют действие или содержимое данной ссылки);
- *логотип компании* (размещенный в левом верхнем углу он является, как правило, точкой перехода на главную страницу);
- *текстовые ссылки* (текст ссылки дает однозначное представление о направлении перехода, кроме того текстовые ссылки удачно вплетаются в основной контент, обеспечивая быстрый переход к дополнительной информации по искомой теме);
- *графические текстовые ссылки (кнопки, ролловеры)* (используются для того чтобы выделить ссылку. Ролловеры – кнопки, меняющие свое состояние при наведении курсора);
- *карты изображения* (отдельные части изображения являются независимыми между собой ссылками и обеспечивают множество переходов с одного и того же рисунка);
- *баннеры* (графические изображения, щелчок по которым переводит на страницу рекламируемого сайта).

Существует ряд рекомендаций и ограничений, позволяющий сделать навигацию по сайту максимально удобной и предсказуемой. Кроме того, для анализа качества системы навигации, используют исследование маршрутов перемещения пользователей по сайту.

Дизайн сайта

Дизайн сайта должен органично дополнять и усиливать общую концепцию.

Основные функции дизайна веб-сайта:

- создание эстетически привлекательного образа на основе общей концепции и фирменного стиля компании;
- обеспечение легкости восприятия информации на странице и акцентирование внимание на ее определенных частях;
- способствование свободной ориентации на сайте посредством создания единого стилистического решения и характерного способа обозначения местоположения посетителя.

Поскольку веб-сайт представляет собой своего рода электронную книгу, то все его странички оформляются в едином стиле, разработанном исходя из целей существования сайта и особенностей представляемой информации.

Инструменты работы с аудиторией (интерактивность)

Для эффективной работы рекламно-информационного сайта существует комплекс средств работы с аудиторией, направленный как на удержание целевой аудитории, так и на сбор информации. Обратная связь позволяет посетителю не только пассивно просматривать информацию, но и быть задействованным в формировании контента веб-сайта.

В качестве средств обратной связи между пользователем и владельцем веб-ресурса могут выступать:

- *почтовая форма* (пользовательская форма, в которой посетитель может изложить суть обращения и предоставить свои персональные данные для получения ответа (чаще всего по электронной почте));
- *гостевая книга* (электронная книга жалоб и предложений, в которой любой желающий может оставить свой отзыв, замечание, и т.п., содержание которого будет доступно всем посетителям);
- интернет-форум (способ организации диалога между человеком, задающим вопрос, и специалистом компании, таким образом, чтобы в обсуждении могли принять участие все заинтересованные лица. Поднятая тема и ответы участников доступны для просмотра всем посетителям сайта в течение долгого времени. Темы диалогов сгруппированы по разделам, что делает их поиск наиболее удобным);
- *чат* (способ организации единовременного общения многих пользователей в режиме реального времени. В отличие от форумов, диалог в чате возможен только между посетителями, одновременно находящимися в комнате чата).

Для изучения пользовательского спроса на тот или иной вид товаров и услуг, выявления и последующего анализа реакции посетителей на какие-либо события, на сайте могут устраиваться всевозможные опросы, анкетирования, голосования, тесты и пр. Электронные бланки, предлагаемые для заполнения посетителям сайта, информация с которых будет передана на сервер для обработки, называются *пользовательскими формами*.

Для получения авторизованного доступа к более подробной или адресной информации, посетителям сайта может быть предложена регистрация на сервисе и составление пользовательского профиля. *Пользовательским профилем* называется совокупность персональных данных, которые указывает о себе посетитель при регистрации (имя, регион проживания, возраст, увлечения и т.п.) и которые впоследствии может изменить путем редактирования.

Таким образом, для создания ощущения сопричастности потребителей к жизни организации, формирования и объединения активной аудитории сайта, необходимо разрабатывать и активно использовать в работе веб-проекта различные формы интерактивности.

УПРАЖНЕНИЕ 4. Жизненный цикл персонального сайта.

1. Продумайте и опишите этапы создания собственного персонального сайта. Представьте в виде подробной схемы жизненный цикл данного информационного продукта.
-

Контрольные вопросы по теме «Веб-сайт компании. Этапы создания веб-сайта».

1. Назовите возможные цели создания Интернет-сайта компании?
2. Перечислите основные этапы создания рекламно-информационного сайта
3. Объясните, как выбор доменного имени сайта влияет на его посещаемость. Чем следует руководствоваться и чего избегать при выборе доменного имени сайта?
4. Что представляет собой система навигации веб-сайта? Каково ее назначение, что входит в состав?
5. Какие основные функции выполняет дизайн сайта?
6. Какие инструменты интерактивности используются для организации обратной связи с посетителями сайта?

ГЛАВА 2. ОСНОВЫ ВИЗУАЛЬНОГО ДИЗАЙНА ВЕБ-РЕСУРСОВ

Страницы сайта проектируются таким образом, чтобы обеспечить ясное представление информации, составляющей основной контент веб-ресурса. При этом, накопленный опыт исследования и проектирования различных информационных продуктов показал, что любое визуальное средство отображения информации должно прорабатываться не только в плане узкой функциональной задачи с целью передачи конкретного содержания, но и как объект эстетического восприятия.

2.1. Основные принципы дизайна

Специалисты в области графического дизайна утверждают, что из множества факторов, определяющих эстетический уровень визуального объекта, особенно существенное влияние на процесс восприятия информации оказывает композиция. Под композицией веб-страницы принято понимать общее расположение, построение всех составных элементов графического макета документа таким образом, чтобы обеспечивались единство и цельность, соподчинение компонентов друг другу и целому.

Всю работу по созданию композиционного решения графического средства представления информации можно разделить на два этапа: 1) компоновку и 2) художественно-композиционное решение графического средства. «Компоновка» подразумевает работу, направленную на достижение функциональной и эргономической целесообразности средства представления информации, «композиция» же направлена на обеспечение высокого художественного уровня этого средства.

Экспериментальные эргономические исследования специалистов в области проектирования различных типов графических средств представления информации позволили сформулировать ряд общих принципов компоновки.

Общие принципы компоновки⁴.

Принцип лаконичности. Графическое средство представления информации должно содержать лишь те элементы, которые необходимы для сообщения наблюдателю существенной информации, точного понимания ее значения или принятия соответствующего оптимального решения.

Принцип обобщения и унификации. 1) Основные формы графического средства не следует излишне дробить, включая в них элементы, несущественные для

⁴ Боумен У. Графическое представление информации. (перевод с англ.) М.: Мир, 1971. – 227 с.

отображаемой информации. 2) В пределах всего комплекса графических средств символы, обозначающие одни и те же объекты или явления должны иметь единое графическое решение.

Принцип акцентирования на основных смысловых элементах. На графических средствах отображения информации следует особо выделять элементы, наиболее существенные с точки зрения передаваемой информации.

Принцип автономности. Части, передающие относительно автономные (самостоятельные) сообщения, следует обособить и четко ограничить от других частей.

Принцип структурности. Каждая автономная часть комплекса графических средств отображения информации, занимающая в общем изложении некоторое центральное, узловое положение, должна иметь четкую, легко запоминающуюся и дифференцирующуюся от других структуру, отражающую характер каждого сообщения.

Принцип стадийности. В зависимости от стадий – последовательных разделов изложения информации – должен выбираться состав сообщений, отображаемых в графической форме на отдельных графических средствах. Этот принцип основан на методах борьбы с иррелевантной (вредной, лишней в конкретном случае) информацией путем ее последовательного восприятия.

Принцип использования привычных ассоциаций и стереотипов. При создании графических средств представления информации должны учитываться устойчивые, привычные ассоциации между символами и обозначаемыми ими объектами и явлениями, а также стереотипные реакции на определенные символы и сигналы.

Художественно-композиционное решение графических средств представления информации должна базироваться на основных принципах графического дизайна (некоторые авторы называют их правилами «коммуникационного» дизайна).

Основные принципы графического дизайна.

Акцентирование.

Акцентирование – подчеркивание особой важности и значимости какого-либо элемента. Некоторые дизайнеры используют иное название для обозначения принципа акцентирования – «фокусная точка».

Фокусной точкой называют визуальный объект, который используется как средство привлечения внимания, точка, от которой пользователь начинает свое путешествие по странице («точка входа»). Фокусная точка – самая важная деталь любого макета. Это та часть изображения, которая обращает на себя внимание, она привлекает глаз зрителя к наиболее важным деталям изображения, где бы ни располагалась.

Применяются и другие термины (в некотором смысле, синонимы), которые описывают данный прием:

- эмфаза⁵;
- визуальный центр.

Существуют два правила, касающиеся использования фокусной точки:

- 1) фокусная точка должна быть. Без нее зритель не поймет, на что смотреть в первую очередь;
- 2) фокусная точка на странице должна быть одна.

Фокусной точкой может быть все что угодно, впечатляющая фотография, иллюстрация, текст или комбинация элементов (рис. 21).

Для правильной расстановки акцентов в рамках одного документа (веб-страницы) требуется проанализировать содержимое, выяснить иерархию элементов в содержимом, выстроить элементы в порядке их важности. Подобное действие позволит избежать акцентирования на всём сразу, а также избежать случайной иерархии.

⁵ Эмфаза – визуальное выделение части текста (изображения) с целью сделать его более заметным, эмоционально окрашенным

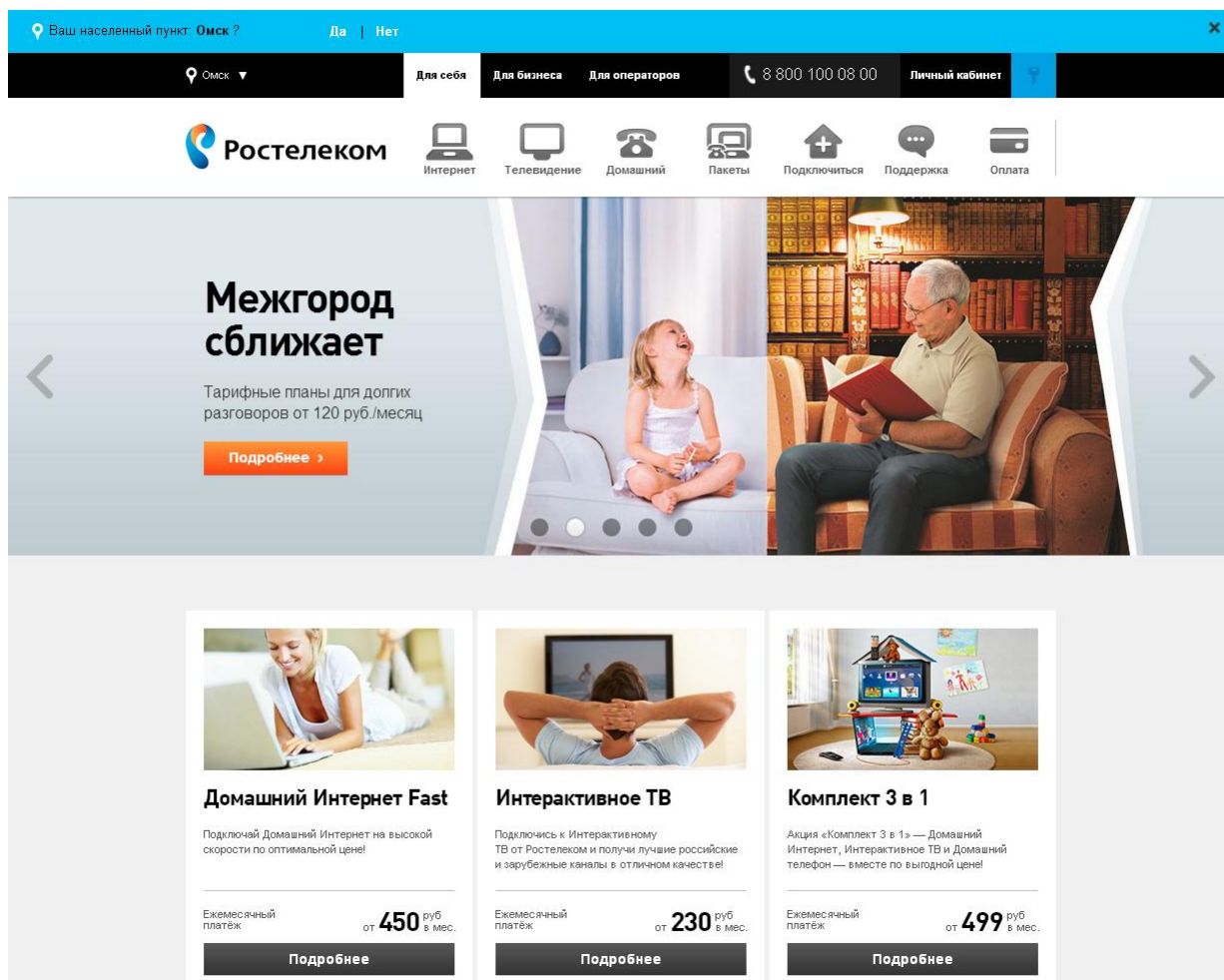


Рис. 21. Большая фотография, крупный шрифт надписи и яркая кнопка акцентируют внимание на информации о новом тарифе (<http://omsk.rt.ru/>)

Контрастирование.

Контрастирование – визуальная дифференциация двух и более элементов. Контрастирование связано с акцентированием, и направлено на удобство восприятия информации на странице веб-сайта. Оно вносит вклад в формирование визуальной иерархии, определяющей важность элементов (рис. 22).

Характеристики элементов, которыми можно манипулировать для повышения контраста:

- цвет;
- размер;
- местоположение;
- тип и толщина шрифта.

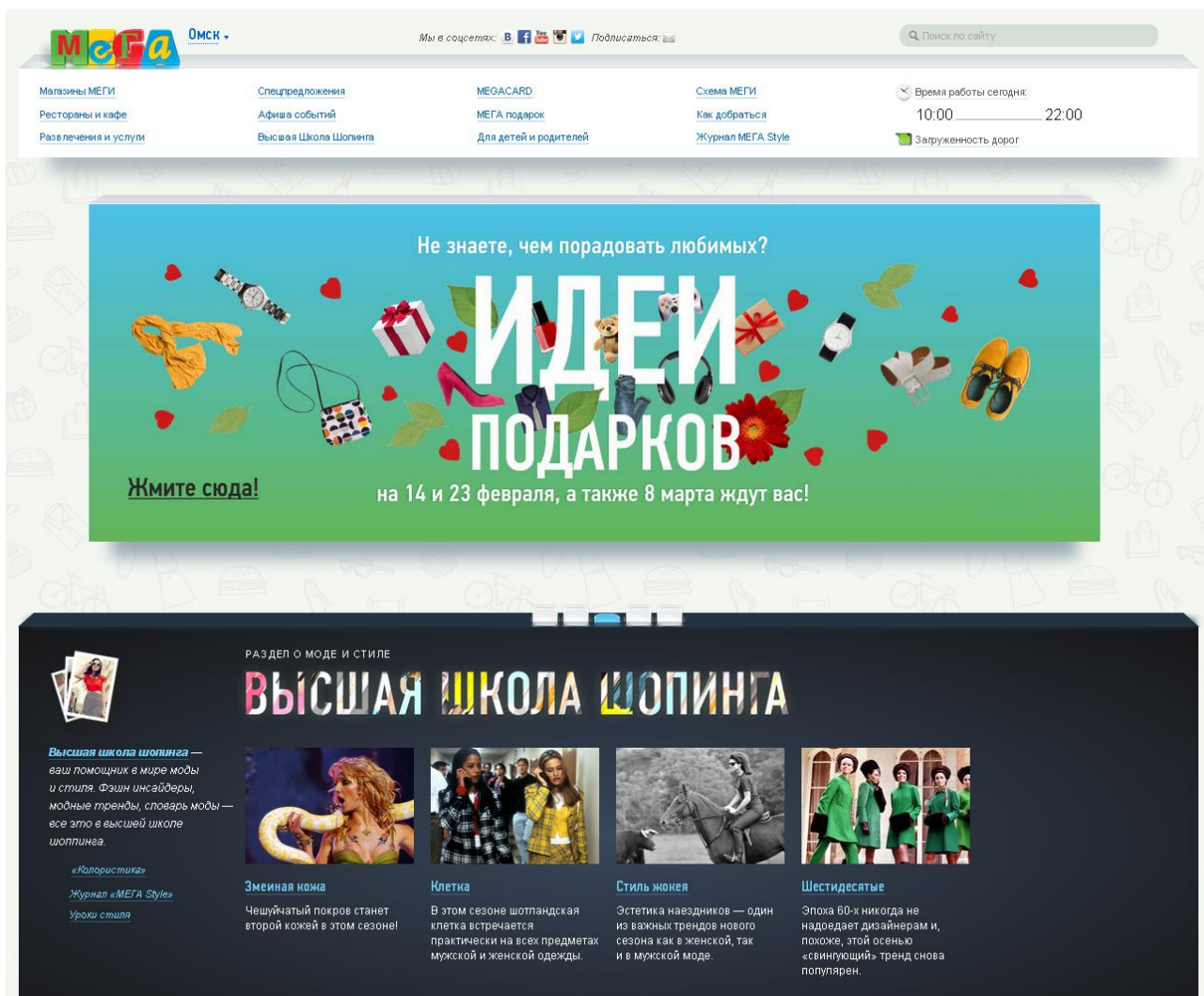


Рис. 22. Контрастирование цветом и размером дифференцирует информационные блоки на странице сети магазинов «Мега» (<http://megamall.ru/>)

Балансировка

Балансировка – распределение визуальной нагрузки в рамках веб-страницы.

Группирование элементов в дизайне создает визуальную нагрузку. Она уравнивается путем использования равнозначной нагрузки «на противоположной чаше весов» (рис.23).

Различают два вида баланса:

1) *симметричный* – левая и правая части относительно некоторой оси являются зеркальными копиями друг друга и несут идентичную визуальную нагрузку (<http://www.mintwheels.com>), (<http://gardenestudio.com.br/>);

2) *асимметричный* – визуальная нагрузка равномерно распределяется по некоторой оси, но отдельные элементы двух составляющих дизайна не являются зеркально одинаковыми (<http://whydbu.dbu.edu>).

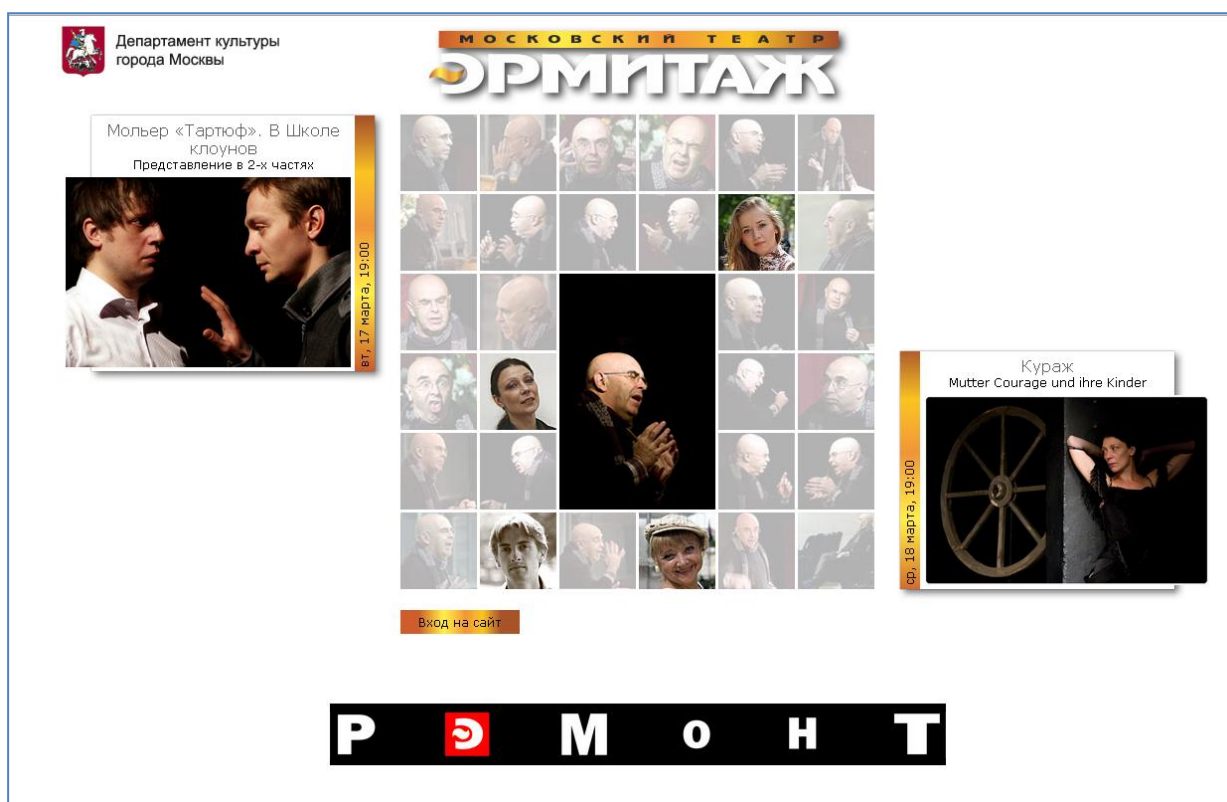


Рис. 23. Визуальная нагрузка левой и правой частей страницы уравновешены (<http://ermitazh.theatre.ru/>)

Встречается также радиальный баланс, когда элементы располагаются вокруг центра кругами.

Выравнивание

Схожие по смыслу элементы часто объединяют в группы. Для визуального обозначения множества элементов как единой группы, их обычно выстраивают в определенном порядке и выравнивают (рис. 24).

Выравнивание – упорядочивание элементов таким образом, чтобы они максимально близко подступали к естественным линиям или границам, которые они образуют. Невыравненные элементы отбиваются от общей группы.

Примерами выравненных элементов могут служить расположенные на одной линии заголовки нескольких столбцов или выстроенные в столбец элементы, левые края которых находящиеся на одной вертикальной линии.

Повторение

Повторение – неоднократное использование в дизайне одних и тех же элементов разными способами. Повторение может выражаться в использовании того же:

- цвета;
- очертаний;
- линий;
- шрифтов;
- изображений;
- общего подхода к стилизации.

Преимущество повторения – предсказуемость, согласованность, постоянство, визуальная связанность.

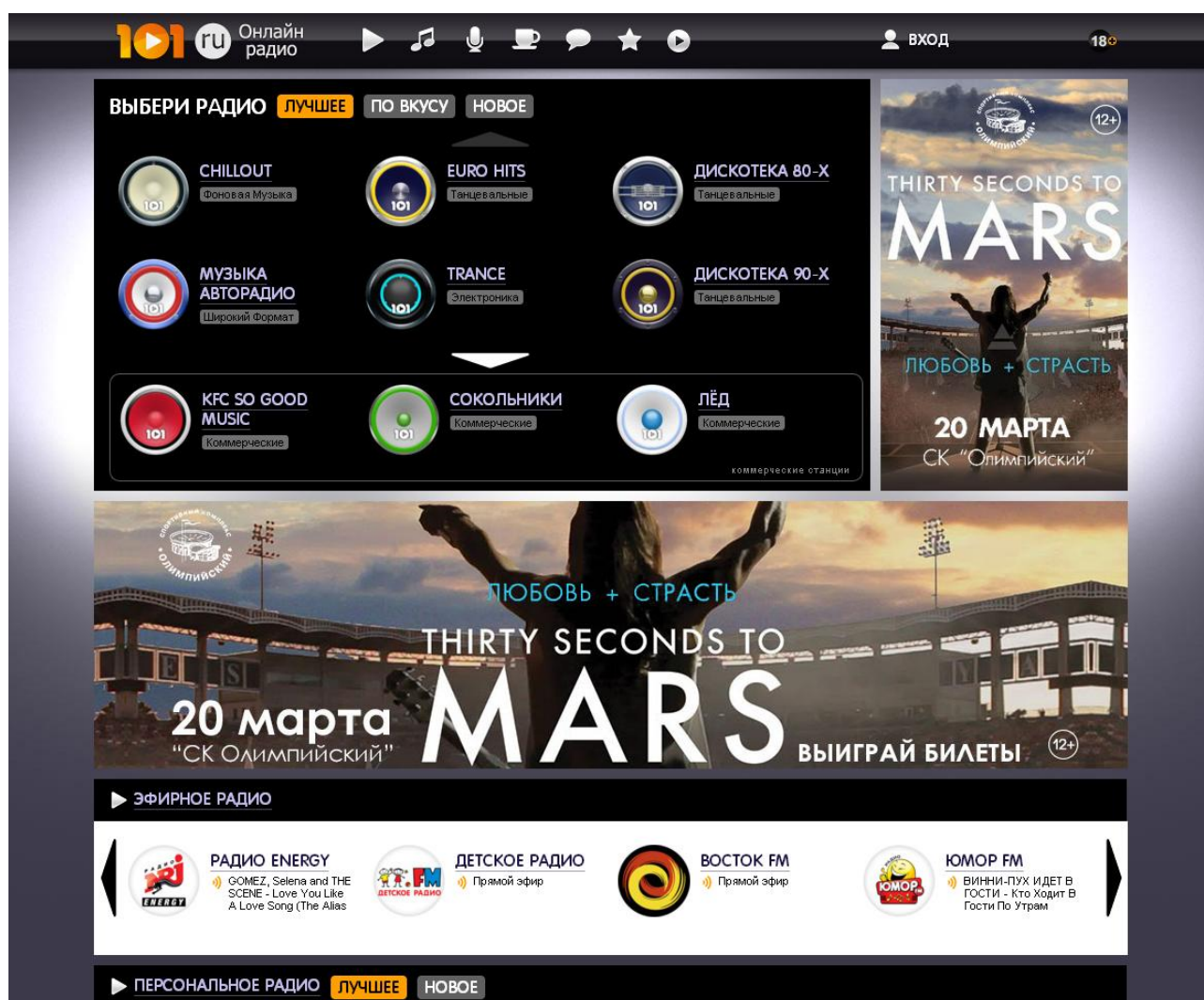


Рис. 24. В обозначении радиостанций повторяется форма. Группы элементов выравнены (<http://101.ru/>)

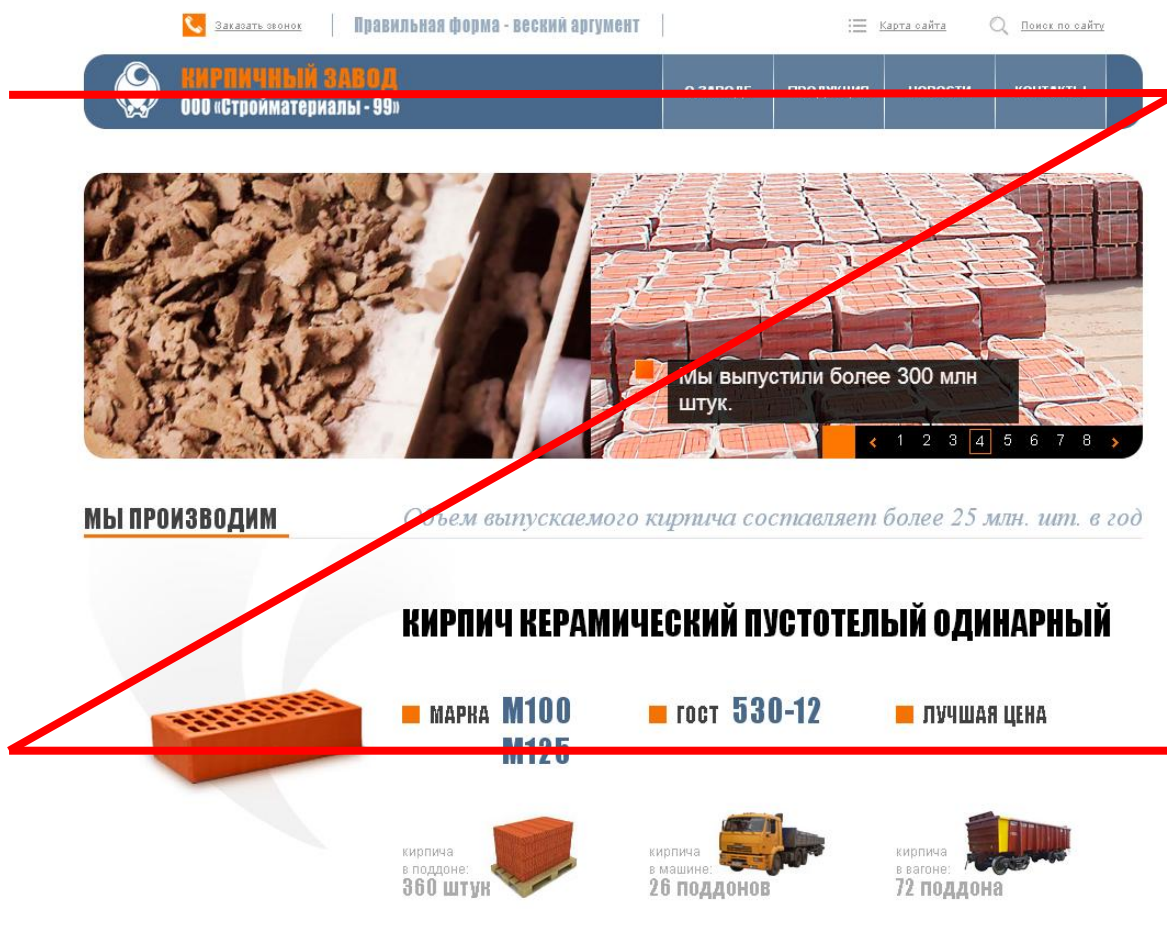
Движение взгляда (обеспечение удобства восприятия)

Люди склонны осматривать те или иные вещи в предсказуемой манере. При визуальном ознакомлении с элементами веб-страницы, глаз следует определенной траектории. Принцип движения основывается на том, что хороший дизайн контролирует и направляет движение глаз по композиции веб-страницы.

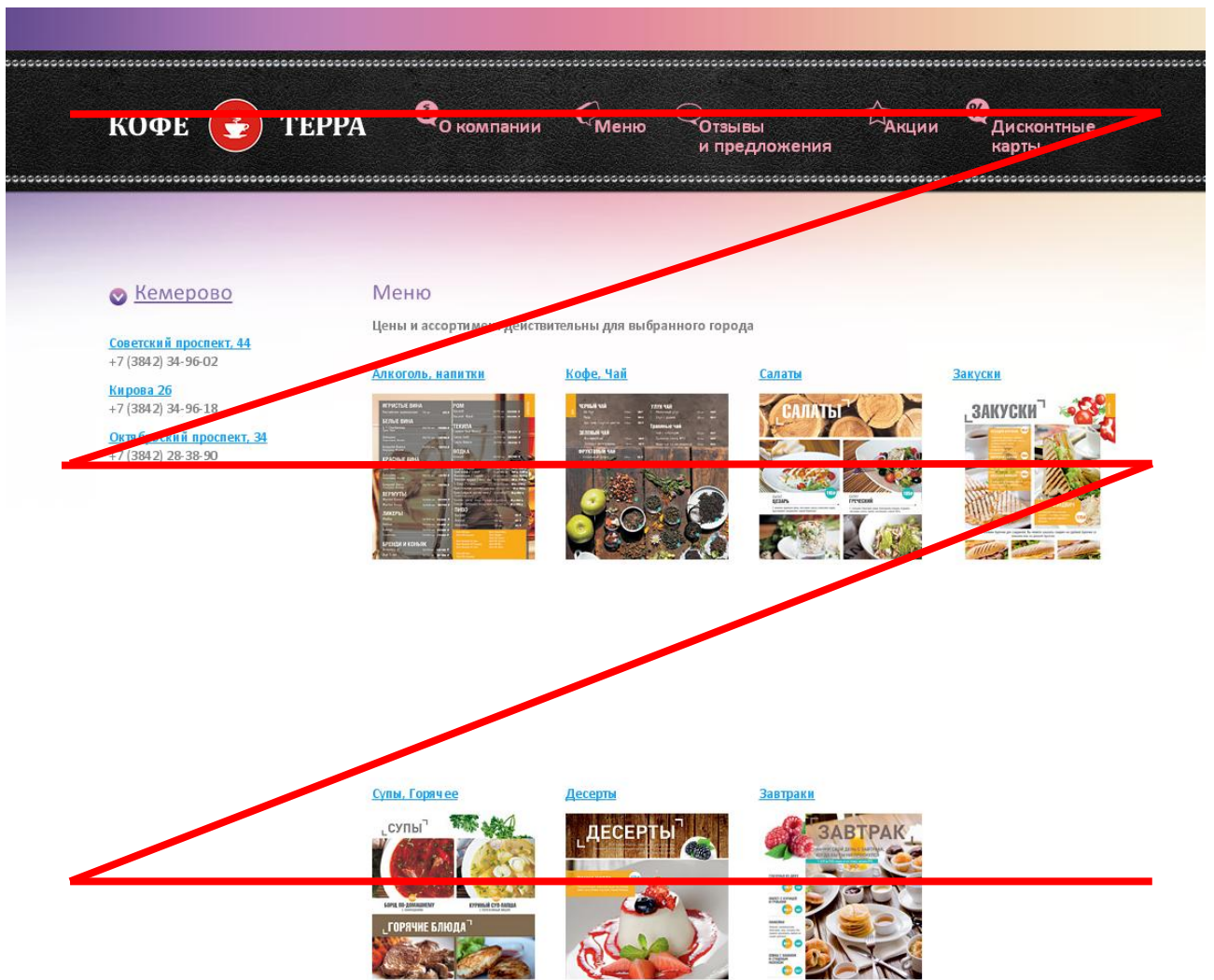
Восприятие страницы динамично – отдельные участки взгляд пробегает быстро, на других задерживается дольше. Взгляд посетителя веб-сайта скользит от одной части композиции к другой, ведомый *непрерывными* линиями или фигурами, а также градациями цвета или фигур.

Динамическое движение (перемещения взгляда по веб-странице), чаще всего, организовывается в соответствии с некоторой схемой. В коммерческом графическом дизайне (дизайне рекламы) есть несколько распространенных схем:

Z-схема (естественная траектория движения взгляда) и подобная ей – «Зиг-заг» (рис. 25 а, б);



а) Траектория движения глаза соответствует букве Z



б) Траектория движения глаза напоминает зиг-заг
 Рис. 25. Страницы, при изучении которых глаз следует по зигзагообразной линии
<http://kofe.terra-group.ru/keмерово/menu>

круговая (приводит взгляд читателя обратно к началу, чтобы послание прочитывалось еще раз) (рис. 26);

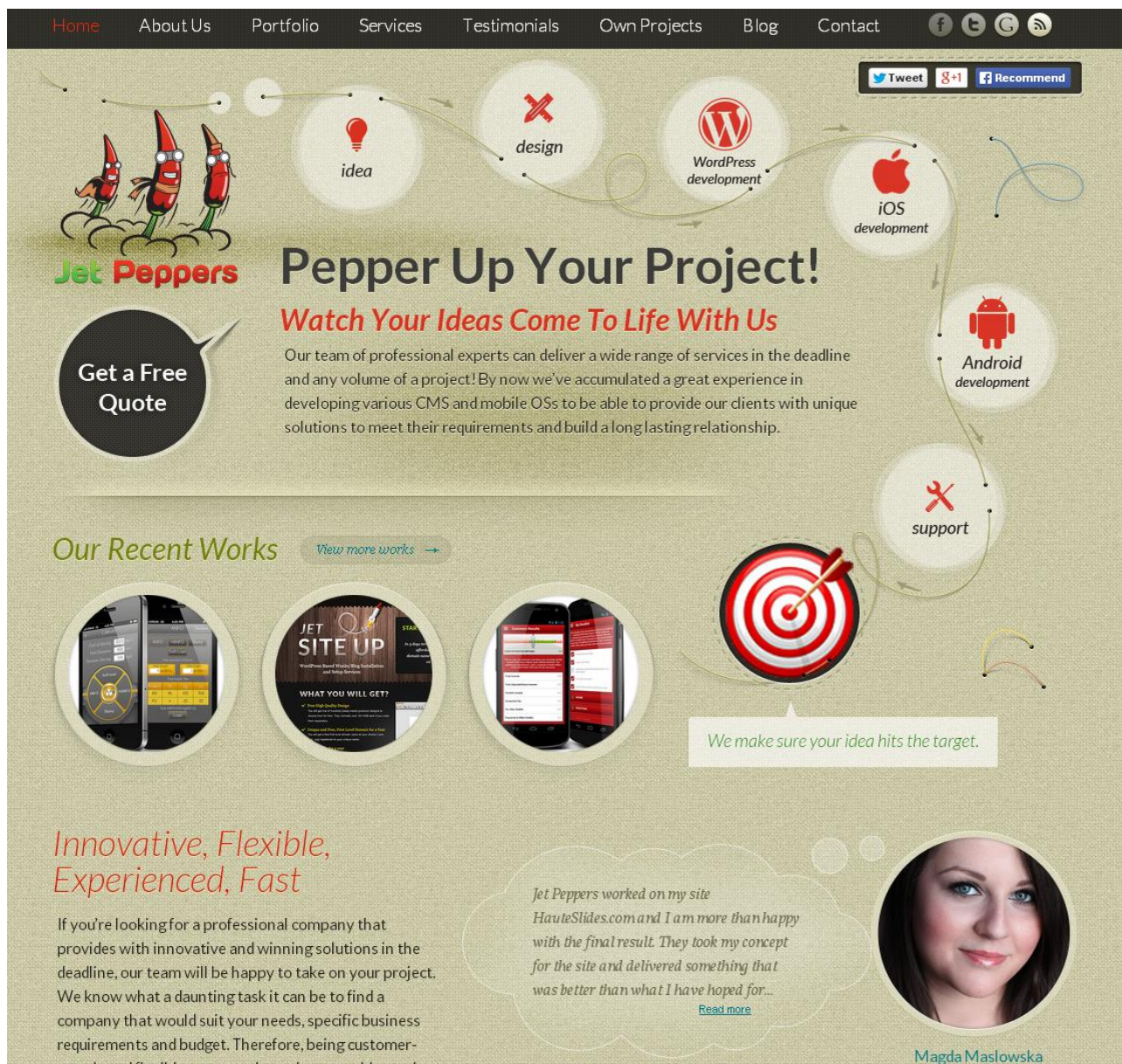


Рис. 26. Страница, при изучении которой глаз следует по круговой линии (<http://jetpeppers.com/>)

F-схема (E) (рис. 27)

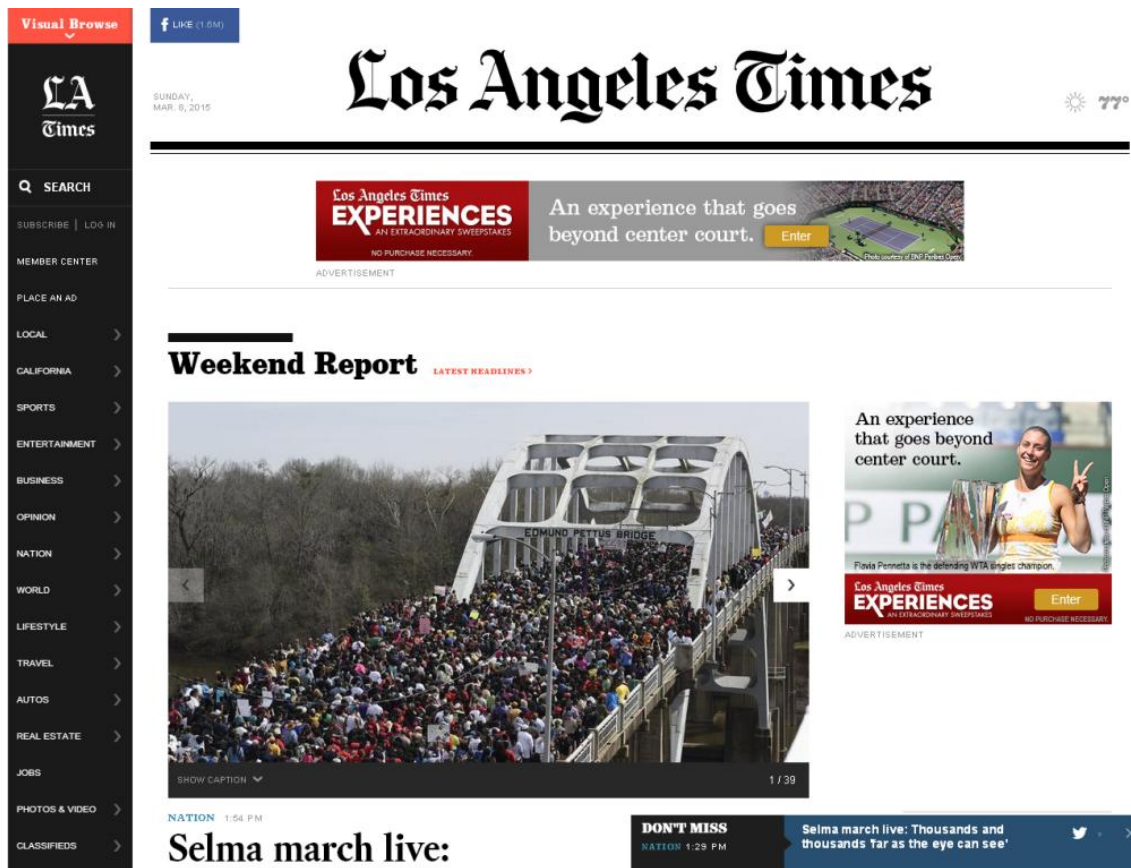


Рис. 27. Страница, при изучении которой траектория движения глаза напоминает букву F

УПРАЖНЕНИЕ 5. Анализ использования основных принципов графического дизайна в оформлении веб-страниц.

1. Найдите в оформлении веб-страниц сайтов различного назначения отражение основных, описанных в тексте параграфа, принципов дизайна.
2. Сделайте вывод о том, соответствует ли дизайн большинства исследованных страниц принципам графического дизайна.
3. Найдите примеры страниц, где композиция направляет движение взгляда пользователя: по Z-схеме, F-схеме, круговой схеме.

Контрольные вопросы по теме «Основные принципы дизайна».

1. Назовите основные принципы графического дизайна, используемые в оформлении веб-страниц.
2. Почему на веб-странице не могут присутствовать две фокусные точки?
3. Назовите основные схемы движения взгляда по веб-странице.

2.2. Верстка веб-страниц

В веб-дизайне, как и в традиционной полиграфии, под версткой понимается процесс размещения текста и иллюстраций в заданном пространстве листа.

При верстке веб-страница рассматривается как набор прямоугольных блоков, которые располагаются в определенном порядке. Слово «порядок» предполагает обращение к одному из основных принципов дизайна – выравниванию. Выравнивание обеспечивает порядок, который, кроме прочего, позволяет быстро найти интересующий нас предмет. Для обеспечения выравнивания, верстку уместно выполнять с применением вспомогательных вертикальных и горизонтальных линий.

Расчерченную горизонтальными и вертикальными линиями поверхность называют *сеткой*. Каркас из линий сетки позволяет задавать скелетную структуру макета документа. По линиям сетки группируют и выравнивают визуальные объекты, текст и отрицательное пространство⁶. Сетка позволяет стабилизировать структуру веб-страницы, облегчает размещение разнообразных данных в плоскости документа. В этом смысле сетка представляет собой логический шаблон для создания сайта.

(ОПАСНОСТЬ: используя сетку, легко забыть о главных и дополнительных элементах дизайна – сетка уравнивает всех. Сетка – инструмент, а не самоцель дизайна)

Для создания сетки, которая станет основой будущего макета, следует определить:

- формат (размер площадки размещения контента веб-страницы);
- тип сетки (регулярная или ассиметричная; колончатая или клеточная).

Формат

Веб-страницы просматриваются пользователями экранов разных размеров и с разными соотношениями сторон. Кроме того, окно браузера не обязательно будет открыто пользователем «во весь экран». Эти обстоятельства заставляют уже на начальном этапе создания сетки определиться (естественно, на основе аргументов), какой из двух вариантов использовать:

- формат, фиксированный по ширине;

⁶ *Положительным* называют пространство, которое занимают объекты и элементы дизайна. *Отрицательное* пространство – это область вокруг объектов.

- гибкий формат (резиновая сетка – используется для адаптивной верстки).

Модульная сетка

Модульная сетка – это набор скрытых направляющих, вдоль которых могут располагаться элементы веб-страницы, модуль – условный блок в конкретной сетке, шаг в ее ритме.

Модульная сетка четко задает структуру макета, описывает размещение блоков в дизайне сайта. Кроме того, она позволяет создать единый стандарт возможного расположения блоков для всех страниц сайта⁷.

Структура модульной сетки в веб-дизайне такова: сетка состоит из юнитов и промежутков между ними (рис. 28). *Юнит* – это единичный компонент, на основе которого формируются колонки. Каждый юнит по умолчанию имеет правый отступ и левый отступ, эти отступы одинаковой ширины. Расстояние между юнитами — сумма правого и левого отступов соседних юнитов. Между крайними юнитами и границами сетки тоже есть отступы. Таким образом, модульную сетку описывают три основных параметра: 1) ширина всей сетки; 2) ширина юнита; и 3) ширина отступов между юнитами.

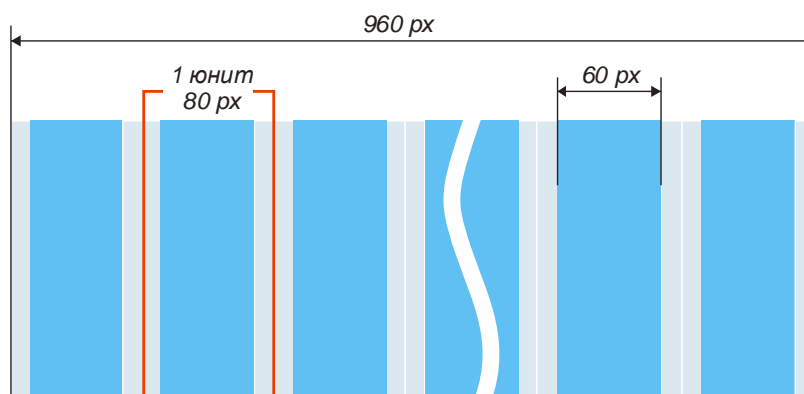


Рис. 28. Структура модульной сетки

Представляемые на веб-странице данные размещают по колонкам. *Колонки* – это группы юнитов, которые объединяются, чтобы создать рабочую зону, подходящую для размещения материала. Большинство текстовых колонок состоят из двух или нескольких юнитов. Например, сетка из двенадцати юнитов может образовывать шесть колонок по два юнита в каждой, либо три колонки по четыре юнита в каждой и т.д.

⁷ <http://modulargrid.org/> — онлайн сервис для генерации модульной сетки. Позволяет создать собственную модульную сетку и скачать ее в формате PNG, в виде паттерна для Фотошопа или прозрачной маски.

В зависимости от количества материала и целей сайта применяют различные типы модульной сетки термины одно-, двух-, трех- или многоколоновые. Рассмотрим эти варианты.

Обычно при создании эскиза применяют схематические обозначения. Например, текст обозначают горизонтальной штриховкой, а изображения – перечеркнутыми прямоугольниками или затемненными блоками.



Одноколонный макет

Одноколончатая модульная сетка чаще всего применяется при публикации большого текста. Такой дизайн отличает минимализм оформления – здесь основное внимание уделяется содержательной части.

В типичной схеме одноколонной модульной сетки обычно присутствуют четыре основных блока:

- заголовок страницы,
- навигация по сайту,
- основной текст,
- контактная и правовая информация (в самом низу страницы).

При большом объеме основного текста, когда высота страницы очень большая, то внизу дублируют блок навигации по сайту или ставят ссылку «Наверх» (рис. 29).

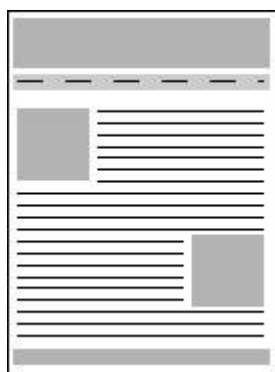


Рис. 29. Одноколонная верстка веб-страницы

Как видно из примера, на странице могут присутствовать рисунки, иллюстрирующие текст, при этом, текст обтекает их по контуру.

Если на сайте планируется размещать много изображений, то удобнее всего применять фиксированный макет. В этом случае изображения можно делать одного размера, с которым они идеально впишутся в макет веб-страницы.

Двухколонный макет

Такая модульная сетка состоит из двух колонок: в одной размещается основной текст, во второй – навигация и другая необходимая информация. Колонка с навигацией может располагаться как слева так и справа (рис.30).

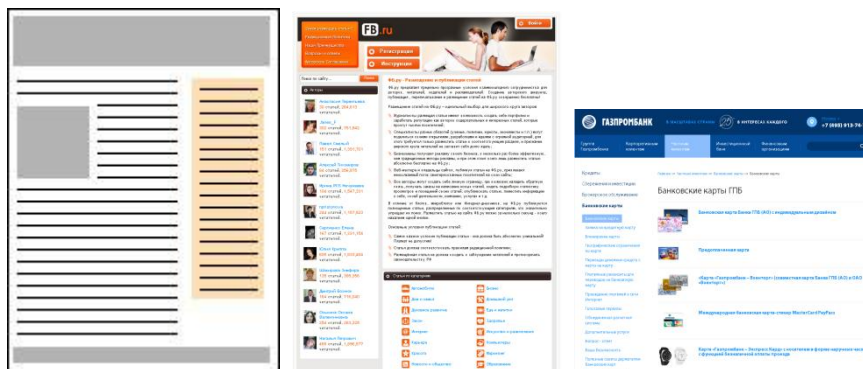


Рис. 30. Двухколонная верстка веб-страницы

Двухколонные сетки очень удобны для создания информационных сайтов. В таком макете все элементы находятся в непосредственной близости: и основной текст и навигация, и ссылки.

Основным недостатком сайтов, сверстаных на основе двухколончатого макета является то, что такие сайты выглядят достаточно однообразно. Однако, оно же и плюс – пользователям удобно работать с сайтом привычного вида.

Трехколонный макет

Чаще всего трехколоночную сетку применяют на главной странице сайта, где пользователю нужно показать много информации. Также такая сетка применяется и на внутренних страницах, если двух колонок не хватает для размещения всей информации (рис. 31).



Рис. 31. Трехколонная верстка веб-страницы

В таком макете в одной колонке размещают навигацию, во второй – основной текст, а в третьей колонке помещают дополнительную информацию (ссылки, рекламу и т. д.).

Многоколонный макет

В настоящее время активно используют многоколонные модульные сетки. Такая сетка позволяет разбить контент веб-страницы на отдельные фрагменты, объединить колонки или отделить различные блоки (рис. 32).

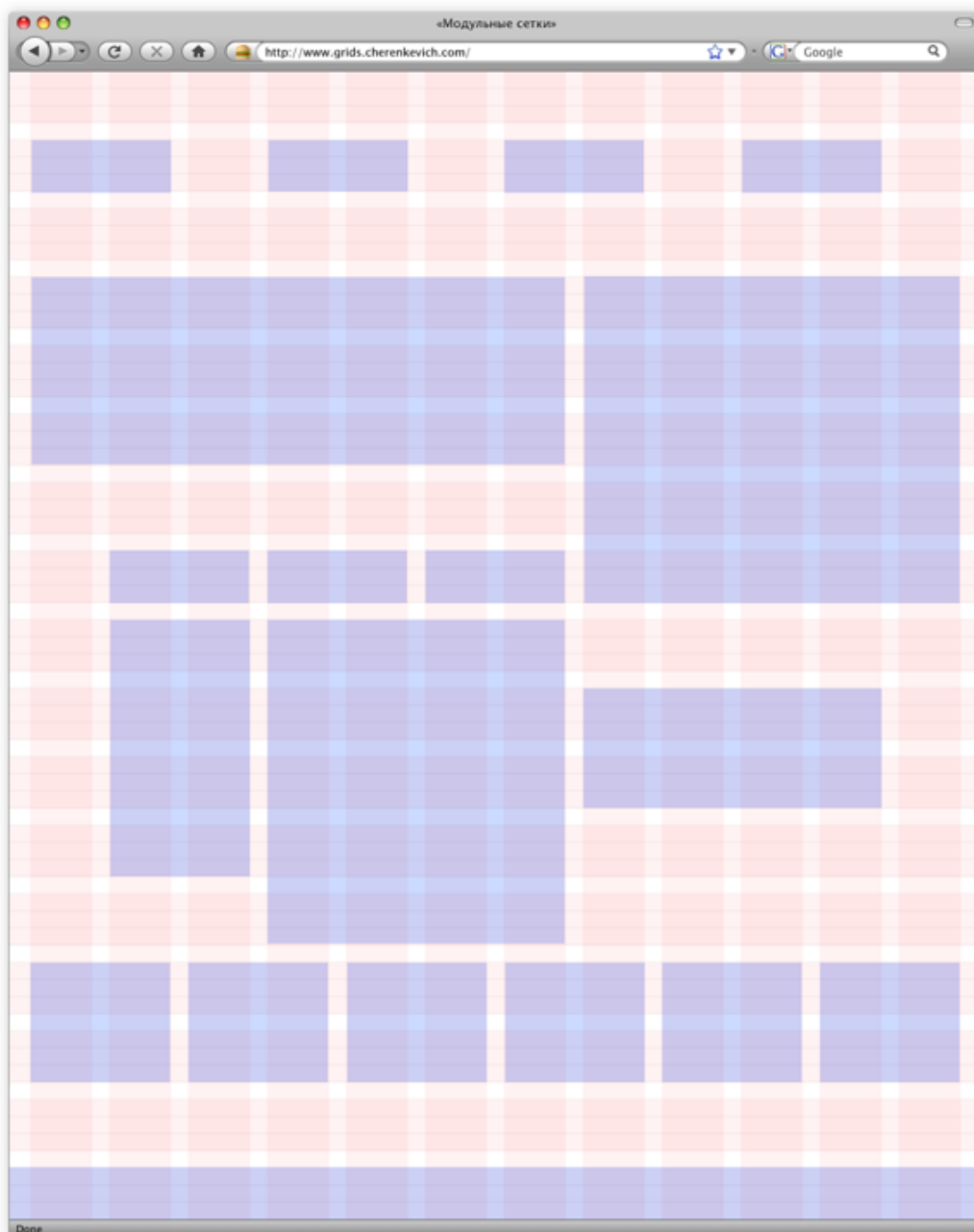


Рис. 32. Многоколонная верстка веб-страницы

Существуют специальные ресурсы, которые генерируют своеобразный css-фреймворк (файл описания параметров блоков), верстальщику остаётся только скачать этот файл, подключить его, и использовать уже готовые блоки нужной ширины.

УПРАЖНЕНИЕ 6. Верстка веб-страниц на основе модульной сетки.

1. Найдите примеры веб-страниц с одно-, двух-, трех- и многоколонной версткой.
 2. С помощью любого онлайн-сервиса, предоставляющего услуги генерации модульной сетки, создайте собственную модульную сетку и на ее основе выполните верстку макета главной страницы или страницы «Мои фотографии» воображаемого персонального сайта.
-

Контрольные вопросы по теме «Верстка веб-страниц».

1. Какой процесс называется версткой сайта?
2. Что представляет собой модульная сетка, для чего она используется?
3. Для каких типов веб-страниц характерны одно-, двух-, трех- и многоколонный макеты?

2.3. Адаптивный дизайн сайтов

Благодаря развитию мобильных технологий, в последние годы все больше и больше людей используют свои гаджеты для доступа в Интернет. Однако, такие особенности мобильных устройств, как небольшой размер дисплея и сенсорный экран делают их неудобными для просмотра обычных веб-сайтов.

По версии исследовательской компании Morgan Stanley Research, число мобильных интернет-пользователей в 2014 году превысит число стационарных пользователей во всем мире, а это означает, что более половины трафика будут генерировать мобильные устройства (мобильные телефоны, смартфоны и коммуникаторы) (рис. 33).

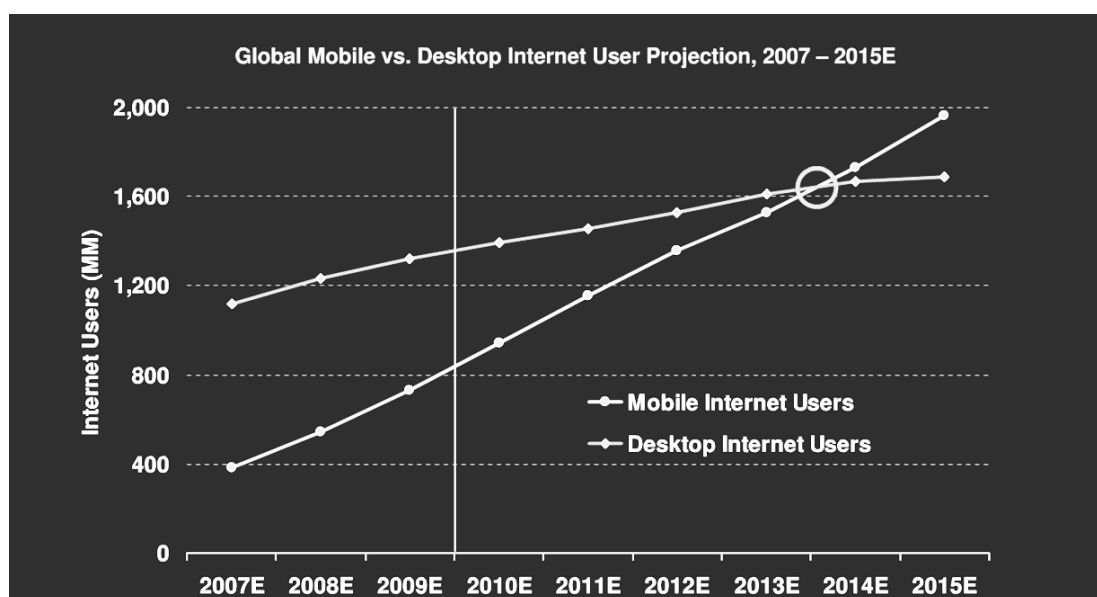


Рис. 33. График изменения числа мобильных интернет-пользователей

По заявлениям некоторых специалистов, данный прогноз говорит, что «если веб-ресурс не будет оптимизирован под мобильные устройства, пользователи будут бежать с такого сайта, как от чумы». Из этого следует – сайты должны адаптироваться к растущему числу мобильных устройств с их относительно небольшими экранами. Решением проблемы стала бы разработка дизайна, адаптирующегося к параметрам устройства вывода, – отзывчивого дизайна (рис. 34).

Впервые термин «отзывчивый веб-дизайн» (Responsive Web Design, RWD) был обозначен в одноименной статье Итана Маркотта, опубликованной 25 мая 2010 года в журнале A List Apart.



Рис. 34. Издание книги Итана Маркотта «Отзывчивый веб-дизайн» предлагается читателям в печатном и электронном виде, адаптированном для всех типов устройств

Отзывчивым, или реагирующим или адаптивным веб-дизайном был назван подход к разработке сайтов, согласно которому ресурс должен быть удобным для просмотра с любого устройства, независимо от размера экрана. Таким устройством может быть: стационарный компьютер, ноутбук, планшет и пр.

Рассмотрим подробнее идеологию (и технологию) адаптивного дизайна.

Современные мобильные устройства обладают широким функционалом и по многим параметрам уже не уступают настольным компьютерам. Однако у них есть своя специфика и особенности использования, которые следует учитывать:

- *маленький экран.* Этот параметр может трактоваться двояко. Во-первых, мобильные устройства имеют меньший размер экрана. У смартфонов этот размер составляет 3-4 дюйма, у планшетов -7-10, тогда

как средний ноутбук имеет размер экрана 15 дюймов, а настольный монитор — 19 и больше. Во-вторых, разрешение экрана смартфонов также не превышает разрешения современных мониторов;

- *низкое быстродействие*. Несмотря на стремительное развитие, смартфоны пока отстают по этому параметру от настольных компьютеров и ноутбуков;
- *меньшая пропускная способность мобильных сетей*. Даже при поддержке 3G она, как правило, не превышает 1 Мбит/сек, а более ранние протоколы имеют еще меньшую скорость передачи данных;
- *способ ввода*. Большинство современных смартфонов и планшетов оснащены сенсорными экранами, которые характеризуются ограниченной точностью позиционирования;
- *неудобство клавиатуры*. Известно, что любая, даже qwerty-клавиатура мобильного устройства, уступает в удобстве самой простой настольной из-за маленького размера клавиш и сложности набора некоторых символов;
- *ограниченная многозадачность*. Работа в сети Интернет с помощью мобильных устройств сопровождается большим количеством ограничений. Например, возможность полноценного перемещения между открытыми вкладками крайне проблематична.

Перечисленные аспекты, и, в особенности, маленький размер дисплея, делают просмотр обычного веб-сайта на мобильном устройстве несколько затруднительным.

Широко распространенным подходом решения этой проблемы является *создание самостоятельной версии сайта для каждого из устройств (PDA-версии от англ. Personal Digital Assistant – персональный цифровой помощник)*. Однако, разнообразие портативных устройств настолько велико, что возникает необходимость создания сразу нескольких версий для мобильных устройств, что экономически нецелесообразно. Кроме того, дублирование информации на различных версиях сайта плохо сказывается на позициях сайта в поисковых системах и вебометрических рейтингах.

Гораздо более перспективным является создание адаптивного веб-сайта, способного приспосабливаться к требуемым параметрам различных устройств. Технология построения сайтов по принципу адаптивного дизайна позволяет создать один сайт, имеющий оптимальное отображение на всех видах устройств, при этом не требующий никаких действий со стороны

пользователя (ручной или автоматический переход на мобильную версию, если разрешение меньше заданного).

Основная идея адаптивного веб-дизайна состоит в том, что сайт должен «реагировать» на устройство, на котором его просматривают. Итан Маркотт, автор книги «Отзывчивый веб-дизайн», полагает, что для создания отзывчивого дизайна требуются три основных компонента:

- гибкий макет на основе сетки (flexible, grid-based layout)
- гибкие изображения (flexible images)
- медиазапросы (media queries), модуль спецификации CSS3.

В более широком плане это может означать следующее:

- *адаптацию шаблона в соответствии с различными размерами экранов*, начиная от широкоэкранных настольных компьютеров до крошечных телефонов;
- *изменение размеров изображения* в соответствии с разрешением экрана;
- *обслуживание низкоскоростных изображений* на мобильных устройствах;
- *упрощение элементов страницы* для использования в мобильных устройствах;
- *скрытие несущественных элементов* на небольших экранах;
- *предоставление больших по размеру и удобных для нажатия пальцами ссылок и кнопок* для мобильных пользователей;
- *выявление и реагирование на мобильные функции*, такие как геолокация и ориентация устройства.

В основу «гибкого макета» положен принцип пропорций (fluid grids). Идея «гибкого» или «текучего» макета состоит в том, что при верстке вместо абсолютных величин используются относительные, т.е. значения задаются в процентах и «em». Так, например, для задания ширины блока в процентах, искомое значение находится по следующей формуле:

ширина блока (%) = ширина блока (px)/ширина макета (px).

Основная идея «гибких изображений» (flexible images) заключается в том, что для изображений следует применять свойство {max-width: 100%}. В этом случае изображения не будут выходить за пределы родительского блока, даже если они превышают размеры блока. Изображение при этом будет

пропорционально уменьшаться. Этот же подход применим к видео, флеш-вставкам и другим объектам.

Медиазапросы (media queries) являются частью стандарта CSS, которая позволяет применять стили на основе информации о разрешении экрана, высоте и ширине браузера, ориентации в пространстве и возможностях устройства. Идея медиазапросов заключается в том, чтобы определить, как размещать информацию, какую информацию показывать, а какую нет, в зависимости от значения всех этих параметров.

Например, при создании media query для окон, ширина которых менее 700 пикселей, можно применить набор пользовательских правил CSS, которые автоматически подстраивают элементы страницы, когда страница выводится на узкие окна браузера.

Вот некоторые наборы правил стандарта CSS media queries, где width — это ширина окна браузера, а device-width — ширина экрана устройства:

- min-width: width (применяется, если ширина окна больше или равна ширине шаблона);
- max-width: width (применяется, если ширина окна меньше или равна ширине шаблона);
- min-device-width: width (применяется при ширине экрана устройства больше или равной ширине шаблона);
- max-device-width: width (применяется при ширине экрана устройства меньше или равной ширине шаблона).

Если экран маленький, необходимо перегруппировать все элементы так, чтобы они разумно уместались на экране, если большой, значит нужно оптимально занять доступное место.

При создании адаптивного сайта следует учитывать, что мобильные устройства, с которых будет просматриваться сайт, в большинстве случаев имеют сенсорный экран. Человеческие пальцы — не самый точный инструмент. Им не хватает аккуратности компьютерной мыши. При разработке тач-интерфейса сайта необходимо это учитывать. Так, ссылки, элементы навигации и пункты меню не следует делать слишком мелкими. Таким образом, следует придерживаться правила: чем важнее выполняемая задача, тем крупнее должна быть зона касания. В руководстве iOS Human Interface Guidelines Apple рекомендует, чтобы размер зоны для нажатия составлял 44×44 пункта. В качестве единицы измерения компания Apple

использует пункты, а не пиксели, поскольку экраны разных устройств имеют разное разрешение. Компания Microsoft также применяет этот подход в руководстве для Windows Phone 7. В частности, предусмотрено, что оптимальный размер тач-зоны (исходя из средних размеров человеческого пальца) должен составлять 9 мм, минимальный – 7 мм, а минимальное расстояние между различными зонами – 2 мм. Существуют и другие руководства для разработчиков, которые рекомендуют примерно такие же размеры, как и Microsoft.

Методология адаптивного дизайна не ограничивается лишь дизайном макета – сам процесс разработки также претерпевает изменения. Если ранее разработчики при создании сайта ориентировались, прежде всего, на стационарные ПК, а уже потом дорабатывали и упрощали его для корректной работы в мобильных браузерах, то сейчас считается целесообразным начинать разработку «от простого к сложному». Таким образом, первостепенной задачей считается воплощение базового функционала для мобильных устройств с последующим развитием дополнительных возможностей для пользователей стационарных компьютеров. Такой подход вынуждает сконцентрироваться на наиболее приоритетных задачах и функциях, приняв ограничения, присущие мобильным устройствам.

УПРАЖНЕНИЕ 7. Адаптивный дизайн сайтов.

1. Выберите для анализа интернет-ресурсы трех типов: новостной, интернет-представительство, сайт-визитка (не менее 3 каждого типа).
2. Откройте выбранные сайты на двух-трех устройствах, имеющих возможность просмотра веб-содержимого: персональном компьютере и смартфоне/планшете. Оцените адаптацию сайтов разного типа к просмотру на маленьком экране. Отметьте:
 - 1) предусмотрена или нет адаптация сайта для просмотра на мобильных устройствах? Отметьте, если адаптация предусмотрена, то это решено существованием двух версий (полная и для мобильных устройств) или адаптивным дизайном (мобильная версия имеет собственный URL, как правило, отличающийся от url полной версии сайта).
 - 2) изменяются размеры рисунков при просмотре на маленьком экране
 - 3) скрывается часть несущественного контента
 - 4) предусмотрена адаптация навигационных элементов для работы на сенсорном экране - изменяются размеры кликабельных областей

(крупные кнопки), увеличенное расстояние между кликабельными элементами)

- 5) происходит перераспределение элементов веб-страницы при вертикальной и горизонтальной ориентации экрана
3. На основании проведенного исследования сделайте вывод – сайты какого типа лучше адаптированы к просмотру на мобильном устройстве? Как полагаете, почему?

Контрольные вопросы по теме «Адаптивный дизайн сайтов».

1. Что подразумевает адаптивный/отзывчивый дизайн сайтов? Чем обусловлена его актуальность?
2. Назовите три основных компонента отзывчивого дизайна (по версии Итана Маркотта).
3. Какие правила следует соблюдать при разработке тач-интерфейса?

2.4. Основные аспекты улучшения качества веб-ресурса. Юзабилити

Одним из ключевых факторов, определяющих качество интернет-ресурса является юзабилити. Понятие «юзабилити» относительно молодое и имеет множество трактовок.

В микроэргономике⁸ понятие «юзабилити» (от англ. usability – способность быть использованным) обозначает итоговый уровень удобства предмета для использования в заявленных целях. В этом случае, термин имеет связь с понятием «эргономичность», но в отличие от последнего, меньше ассоциируется с технической эстетикой, с внешним видом и более привязан к утилитарности «юзабельного» объекта.

При разработке пользовательских интерфейсов словом «юзабилити» обозначают общую концепцию их удобства при использовании программного обеспечения, логичность и простоту в расположении элементов управления.

Иными словами, *юзабилити* рассматривают как эргономическую характеристику того, насколько продукт может быть эффективно, экономично и с удовольствием использован определенными пользователями для достижения поставленных целей в заданном контексте использования.

Международный стандарт ISO 9241-11 определяет *юзабилити* как «Свойство продукции, при наличии которого установленный пользователь может применить продукцию в определенных условиях использования для достижения установленных целей с необходимой результативностью, эффективностью и удовлетворенностью».

С точки зрения веб-дизайна, юзабилити, как характеристика сайта, включает в себя удобство и простоту интерфейса, навигации, типографику (для удобства восприятия текстовой информации), цветовую схему и другие аспекты коммуникации.

Причины по которым необходимо уделять внимание юзабилити следующие:

- во-первых, чем удобнее пользоваться услугами интернет-ресурса, тем чаще пользователи выполняют полезные для бизнеса действия;
- во-вторых, повышается лояльность к ресурсу, что увеличивает конкурентоспособность;

⁸ *Эргономика* – это наука о приспособлении должностных обязанностей, рабочих мест, предметов и объектов труда, а также компьютерных программ для наиболее безопасного и эффективного труда работника. *Микроэргономика*, в свою очередь, – область эргономики, рассматривающая дизайн интерфейсов между человеком и др. элементами системы, включая интерфейсы человек — программное обеспечение.

- в-третьих, такие поисковые машины как Google и Яндекс и др. косвенно анализируют юзабилити сайтов (сравнивая количество рекламы, время пребывания пользователя на сайте и контрастность текстов) и поднимают их позиции при выдаче.

Важно отметить, что юзабилити решает вопросы именно удержания пользователя на ресурсе, а не его привлечения. Юзабилити имеет важное значение для повышения показателя (коэффициента) конверсии в электронной коммерции. *Конверсией* называют соотношение всех посетителей интернет-ресурса к числу тех, кто совершил полезное действие. Полезное действие здесь – это выполнение пользователем задачи, для решения которой он воспользовался интернет-ресурсом. Например, для интернет магазина полезным действием является покупка, для промо-сайта – регистрация промо-кода и т.д.

Для того, чтобы понять насколько удобно людям использовать некоторый искусственный объект (веб-сайт) для его предполагаемого применения, разработчики проводят юзабилити-тестирование. Юзабилити-тестирование помогает ответить на следующие вопросы:

- Понимают ли пользователи основное назначение сайта?
- Могут ли пользователи найти на сайте нужную информацию или услуги?
- Насколько простым является для них использование этих услуг?
- Достигаются ли цели, которые определили владельцы сайта?

Каждый тип интернет-ресурса и каждый тип страниц акцентирует внимание на определенных аспектах юзабилити.

Для главных страниц интернет-магазинов, например, ключевым фактором является *навигация по сайту*: от того, насколько быстро пользователь найдет необходимый товар, будет зависеть количество продаж. При этом, дизайн страниц магазина, продающего одну категорию товаров, может значительно отличаться от магазина, являющегося гипермаркетом.

Для страниц оплаты товара или оформления доставки, задачей юзабилити будет обеспечение простоты заполнения пользовательских форм и оперативный, интуитивно понятный доступ к элементам, где размещены ответы на возникающие в данной ситуации вопросы.

Таким образом, требования юзабилити каждого интернет-ресурса определяются его назначением.

Средства и технологии выявления проблем юзабилити интернет-ресурса

Существующие на данный момент технологии позволяют точно отслеживать статистику и поведение пользователей. Это помогает выявить недостатки дизайна, придумать более эффективное решение и протестировать его. Эффективными средствами для сбора данных по пользованию сайтом являются Google Analytics и Яндекс.Метрика. Оба средства являются подключаемыми к сайту модулями, и предоставляют обширные возможности для сбора данных.

Для детального анализа юзабилити отдельной страницы, сервисы предоставляют следующие инструменты.

Карта кликов. На карте кликов соответствующим цветом помечаются места, куда часто нажимают пользователи. С ее помощью можно отследить правильно ли пользователи понимают кликабельные элементы страницы и нажимают ли они туда, куда нужно, для эффективной работы сервиса.

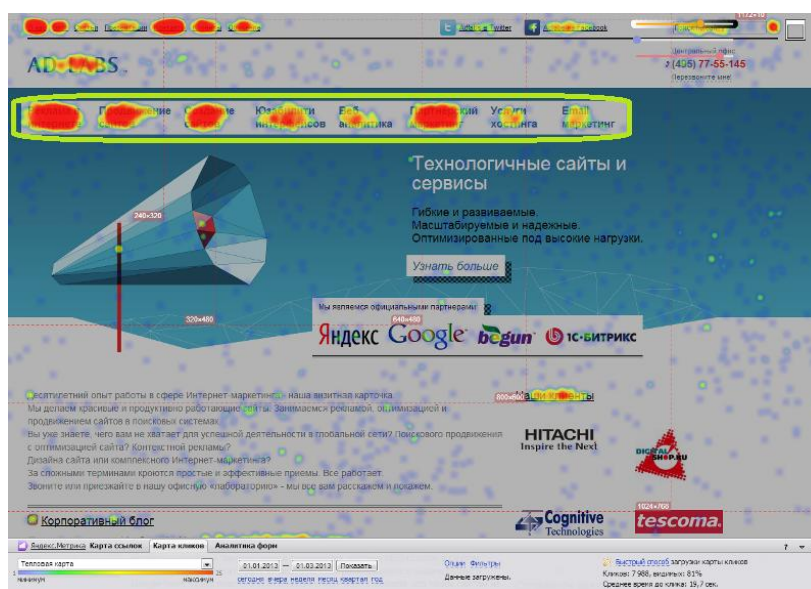


Рис 35. Пример карты кликов

Карта скроллинга. Карта скроллинга позволяет понять, какая часть страницы более интересна посетителям. Работает этот инструмент следующим образом: на исследуемую страницу сайта накладывается тепловая карта, при этом каждой части страницы помечается цветом, соответствующим среднему времени просмотра данного участка страницы и количеству просмотров. Карта скроллинга поможет подобрать оптимальную длину страницы, и подскажет, в какой части страницы необходимо расположить наиболее важную информацию.

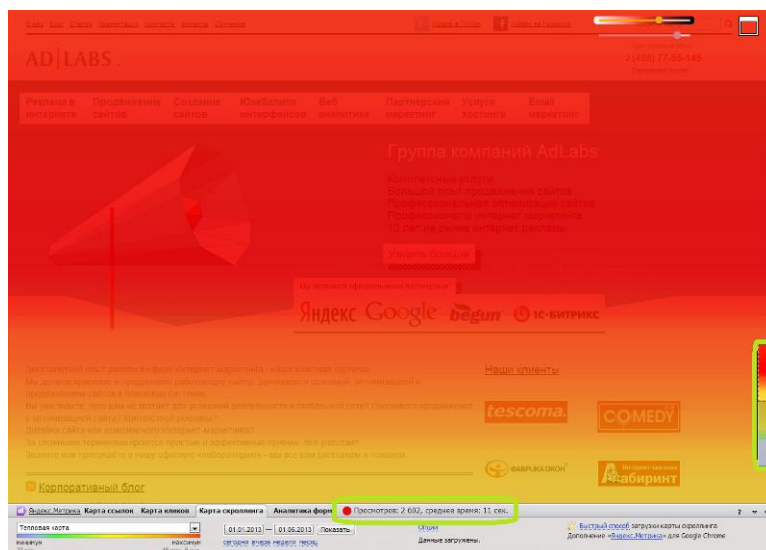


Рис 36. Пример карты скроллинга

Вебвизор. Данный инструмент присутствует в Яндекс.Метрике. Вебвизор позволяет просматривать видеозапись посещения, в том числе передвижения курсора, клики и прочее. Такие данные позволяют увидеть поведение реальных посетителей на основе следа от движения курсора мыши.

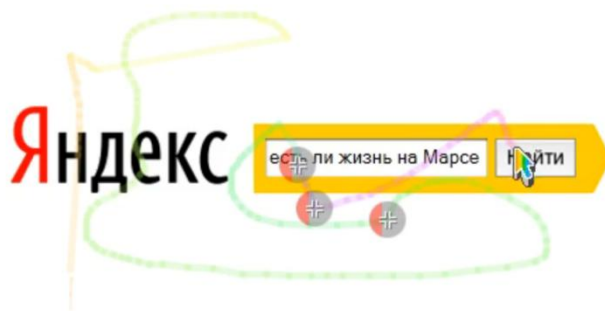


Рис 37. Фрагмент из промо-ролика Яндекс, демонстрирующий работу Вебвизора

Eye tracking. Данная технология представляет собой программно-аппаратный комплекс для слежения за взглядом пользователя. Так как пользователь не всегда передвигает курсор вслед за взглядом, то данные вебвизора не позволяют выявить в какой последовательности он изучает веб-страницу и удалось ли дизайнеру решить задачу управления вниманием посетителя.

Отслеживание взгляда является наиболее эффективным инструментом анализа поведения пользователя. Актуальность технологии обеспечивается тем, что текущие тенденции развития управления техникой сводятся к управлению пальцами и жестами (т.е. без курсора). Результатом работы комплекса Eye tracking является тепловая карта, демонстрирующая частоту и продолжительность взгляда, а так же траекторию его движения. Отслеживание траектории взгляда значительно важнее отслеживания того, куда пользователь направляет курсор, так как чтобы направить его, пользователь должен вначале что-то увидеть.



Рис 38. Аппаратная составляющая комплекса Eye tracking

Основными параметрами при анализе результатов работы пользователя с веб-страницей являются:

- *скорость просмотра*. Если на сайте есть очень важный текст, но он большой и посетители его быстро проматывают, значит, они его не читают. Нужно проанализировать, за какое время просматриваются другие страницы, сравнить, попробовать прочитать этот текст. Блоки с большим объемом информации необходимо делать максимально читабельными. Большие тексты разбивать на абзацы с подзаголовками. Убедится, что оптимально подобран шрифт, кегль, интерлиньяж и прочие параметры. Если есть потребность в том, чтобы пользователь обязательно прочитал информацию (например, лицензионное

соглашение), то ее стоит выводить поэтапно, снабжая краткими тезисами.

- *фокусы внимания.* Грамотное управление вниманием пользователя подразумевает использование объектов, которые привлекают к себе взгляд. Если этого не происходит, пользователь, например, не видит цену, или время работы магазина, то это повод сделать их заметнее.
- *Места кликов и клики по некликабельным элементам и областям.* Зачастую пользователи ведут себя совсем не так, как было задумано автором и некоторые элементы ставят пользователя в тупик. Как правило это касается элементов навигации и управления интернет-ресурса. Например, неочевидная кнопка, или наоборот, кнопка, которая является всего лишь частью изображения.

Помимо анализа отдельных страниц Яндекс.Метрика и Analytics позволяют узнать с каких устройств и браузеров пользователи посещали интернет-ресурс. Эти данные позволяют создать ресурс оптимизированный под устройства, предпочитаемые представителями реальной аудитории сайта.

Помимо описанных выше, подключаемых к сайту модулей, существуют онлайн инструменты юзабилити-тестирования. Они помогают выяснить, насколько продукт соответствует ожиданиям пользователей, и в то же время, экономят время и средства на проведение исследований. Использовать эти сервисы можно даже не имея рабочего сайта, загрузив только макеты будущих страниц. Кратко рассмотрим некоторые сервисы.

Userfly. Userfly является браузерным онлайн-инструментом. Это одно из самых распространенных и доступных средств анализа. Когда пользователи посещают сайт, фиксируется как пользователь водил мышью, что нажимал, куда переходил. Сервис доступен по адресу: [http:// userfly.com/](http://userfly.com/).

UsabilityHub. Данный сервис предоставляет три инструмента: Navflow, Fivesecondtest, ClickTest. Первый позволяет анализировать, насколько легко пользователи ориентируются на странице сайта, второй помогает выявлять элементы дизайна, привлекающие наибольшее внимание посетителей, а третий создает карту кликов (heatmap), показывающую области веб-страницы, на которые пользователи кликают чаще всего. Особенностью приложения является возможность выбора языка тестирования, причем проект на русском языке, например, можно показывать только русскоязычным пользователям. Сервис доступен по адресу: <https://usabilityhub.com/>.

Feng-GUI. Данный сервис имитирует взгляд пользователя в течение первых 5 секунд воздействия визуального эффекта. Это приложение создает карту движения глаз по странице (heatmap) на основе алгоритма, который предсказывает, на что, скорее всего, будет смотреть реальный человек. На главной странице сайта предусмотрена возможность добавить скриншот (мгновенный снимок экрана) веб-страницы и увидеть вероятные области повышенного внимания посетителя. Сервис доступен по адресу: <http://www.feng-gui.com/>.

Optimal Workshop. На сайте OptimalWorkshop можно оптимизировать сайт, используя три разных инструмента:

Optimalsort – это инструмент, который помогает организовать структуру сайта с помощью метода сортировки карточек.

Treejack – приложение для тестирования многоуровневой информационной архитектуры сайта.

Salkmark – предназначен для тестирования юзабилити веб-страниц. Он помогает понять, насколько просто (или сложно) пользователям сайта находить нужную информацию. Чтобы начать работу, требуется загрузить скриншот веб-страницы и поставить перед пользователями какую-либо задачу. Salkmark накапливает ответы и выдает результаты тестирования в виде тепловой карты кликов, а также сообщает среднее время, необходимое для выполнения каждой задачи. В сервисе также присутствует русский язык, что позволяет тестировать соответствующие макеты. Сервис доступен по адресу: <https://www.optimalworkshop.com/>.

Помимо специализированных сервисов, в сети Интернет есть специальные сообщества, где можно продемонстрировать свой макет, или обсудить макеты других участников сообщества. Примеры подобных сообществ: SitePolice, Revision.ru, Russiancreators.ru, Behance, Dribbble, ConceptFeedback.

УПРАЖНЕНИЕ 8. Анализ юзабилити веб-ресурса.

1. Имитируя обращение к услугам электронной регистратуры медицинского учреждения, проанализируйте удобство пользования сайтом «Электронная регистратура Омского здравоохранения» (<http://omskzdrav.ru/>).
2. Оцените юзабилити интернет-ресурса, составьте перечень предложений по улучшению юзабилити данного сайта.

Контрольные вопросы по теме «Основные аспекты улучшения качества веб-ресурса. Юзабилити».

1. Что означает термин «юзабилити»?
2. Для чего веб-разработчики проводят юзабилити-тестирование сайтов?
3. Какие средства используют для анализа юзабилити веб-сайта?

2.5. Основные тенденции и тренды современного веб-дизайна

Под словом «тенденция», как правило, понимают направление, в котором идет развитие некоего явления. Тенденция – это стремление, намерение, свойственное кому-либо или чему-либо; направленность во взглядах или действиях. Устойчивая во времени тенденция называется *трендом*. Тренд выражает закономерность, которая характеризует общую долгосрочную тенденцию в изменениях показателей временного ряда.

В экономике «тренд» – это отрезок числового ряда на котором разница начальных и конечных значений ряда существенно больше колебаний значений внутри отрезка. (Пример графика) Тренд является одним из инструментов анализа, позволяющим стратегически прогнозировать поведение чего-либо.

Поскольку понятие «тренд» имеет математические и экономические корни, можно говорить о ряде общих правил определения силы тренда:

- чем дольше по времени сохраняется тренд, тем он сильнее, однако у него есть предел;
- чем линия тренда круче и быстрее, тем тренд сильнее;
- продолжительный пологий тренд имеет большие шансы на свое продолжение;
- очень крутой тренд может также круто перевернуться;
- любой тренд со временем ослабевает, однако вероятность продолжения тренда в любой его точке выше вероятности его разворота.

Ничто не стоит на месте, и современные технологии движутся в своем развитии весьма стремительно. Сфера медиаиндустрии очень чутко реагирует на технические новшества, и весьма тщательно отслеживает любые «революционные» изменения. В этой связи, специалисты веб-дизайна пытаются заглянуть в будущее и спрогнозировать, какие тренды станут главенствовать в веб-дизайне в предстоящем году.

Чтобы узнать, что будет востребовано в веб-дизайне в ближайший период, принято анализировать тенденции минувшего года. В анализе присутствуют, как правило, две составляющие: одна – это тренды, диктуемые развитием современных мобильных технологий и дизайнерской мысли (в частности, присутствие сильного тренда адаптивного веб-дизайна обусловлено стремительным и массовым развитием мобильного интернета); а вторая – это

модные предпочтения, комбинация из которых позволяет дизайнерам создавать уникальные и неповторимые вещи.

В конце каждого года, лучшие специалисты в области веб-технологий публикуют свои прогнозы относительно тенденций года наступающего. Ниже приведена подборка актуальных трендов веб-дизайна на 2015 года, сформированная на основе анализа авторских материалов. Итак, в ближайшем будущем будут актуальны:

– в области технологических решений:

Адаптивный веб-дизайн. Технический бум всевозможных мобильных и не только устройств жестко призывает веб-дизайнеров делать сайты, адаптированные под всевозможные разрешения, будь то небольшой экран карманного смартфона, средний размер планшета или огромная диагональ современного монитора.

CSS3 эффекты. Возможности CSS3 технологий позволяют веб-дизайнерам творить чудеса, ограниченные лишь размахом их фантазии. Всевозможные движения, появления, трансформации контента, сложные перестроения – все это делает сайты интересными и уникальными.

Параллакс скроллинг. Взаимодействие различных элементов сайта, движущихся с разной скоростью при вертикальной прокрутке страницы, создает очень интересный эффект, который и называется параллакс скроллинг. Возможно, уходящий тренд: Смысл, который последний год все вкладывают в параллакс, — создать «вау-эффект». В итоге каждый второй сайт с параллаксом, и никакого «вау» пользователь уже не чувствует. Сохранится как инструмент интерактивного сторителлинга там, где это уместно.

Видео бэкграунд (Фоновое видео, встроенное на базе HTML5). Большие фоновые видео – еще одна новомодная тенденция. Эмоциональное воздействие полноэкранного видео при знакомстве с сайтом может стать главным решающим фактором и сделать его запоминающимся, так как имеет сильный презентационный эффект. Отличная альтернатива фотографиями и адекватный способ показать, например, технологический процесс без длинных описаний.

SVG-графика. Диапазон разрешений разных устройств заставляет делать как можно больше графики в векторном формате (особенно актуально для иконок, логотипов, маскирующих фигур).

– в области навигации и организации перемещений по странице:

Одностраничные сайты. Одностраничные сайты имеют неоспоримое преимущество в том, что нет необходимости переходить на следующую страницу. Все манипуляции осуществляются в пределах одной страницы. К нужному контенту пользователь попадает либо с помощью мягкого скроллинга, либо в модальных окнах, либо с помощью подгружающегося контента.

Прокрутка (вертикальный и горизонтальный скроллинг). Вертикальный скроллинг. Этот тренд опять таки породили особенности технических устройств, когда оптимально прокручивать контент движением снизу вверх по экрану. А вариации с постраничным вертикальным скроллингом смотрятся очень стильно и современно. <http://www.activate.co.uk/>. Смещение акцентов на мобильное пользование вернуло на сцену и горизонтальный скроллинг, который легко делается свайпом⁹ на планшете/смартфоне. Задача дизайнеров в том, чтобы не затруднять пользование с десктопа, дать подсказку пользователям и предложить вариант навигации, адекватный свайпу (скроллинг колеса мыши – не самый удачный вариант для горизонтальной прокрутки).

Фиксированная навигация. Суть этого тренда в том, что при прокрутке страницы, сверху жестко фиксируется верхнее меню навигации. Это сделано опять-таки в угоду мобильным девайсам. Встречается также и фиксированное нижнее меню.

Боковое сворачивающееся меню. Для бокового свернутого меню уже есть соответствующий узнаваемый значок в виде трех параллельных горизонтальных полосок. В мобильных версиях этот значок закреплен вверху экрана, нажимая на который становится доступным развернутое меню.

– в оформлении внешнего вида:

Плоский дизайн. Плоский дизайн (Flat UI) – это ультра минималистический подход к содержимому, когда избавляются абсолютно ото всего лишнего (объемы, шумы, градиенты и прочие украшения), оставляя только плоские цвета и интуитивно понятные визуальные элементы. В общем, только суть и ничего лишнего.

⁹ *Свайп* (от англ. *swipe* – проводить не отрывая, скользить) – это специальный жест, когда вы кладете палец на экран, а затем ведете его в каком либо направлении не отрывая от экрана, после чего убираете его с экрана.

Нескучная типографика. Типографика прочно обосновалась среди современных трендов веб-дизайна. Если раньше акцент делался на красоту и разнообразие изысканных шрифтов, то теперь добавились еще и эксперименты с геометрией, когда текст или слово вписываются в пространство дизайна и напрямую взаимодействуют с другими элементами.
<http://labubbly.com/>

Модульные конструкции. Очень удачное и интересное решение, когда контент представлен в виде четко локализованных и плотно скомпонованных модульных конструкций или карточек с единообразным оформлением.
<http://silktricky.com/>

Большое фото на заднем фоне. Красивая большая картинка на бэкграунде стала уже устоявшимся трендом минувших лет. Это смотрится броско, эффектно и презентабельно, а также очень активно воздействует на зрителя. Раньше от этого удерживал только слабый интернет-канал у большинства пользователей. Теперь же обратной дороги к крохотными миниатюрам просто нет.

Фотографии с уникальными углами обзора. Этот креативный подход позволяет проиллюстрировать материал одним из самых оригинальных способов.

Анимированные иконки. Тренд, относительно недавно вошедший в моду. Будет жить в 2015-ом.

Простые цветовые схемы. Данная тенденция предполагает использование в дизайне одного, двух или трех чистых простых монохромных цветов в сочетании с классическим черным или белым для остальных элементов дизайна. Своего рода цветная вариация минимализма.

Эффекты «объема» при плоском дизайне. Внешний вид большинства элементов останется «плоским», будет сделан упор на их реалистичной анимации.

Контурная иллюстрация. Сегодня глобальные эстетические предпочтения тяготеют к простоте, а контурные рисунки и контурная иллюстрация в целом становятся все более популярными (рис.).

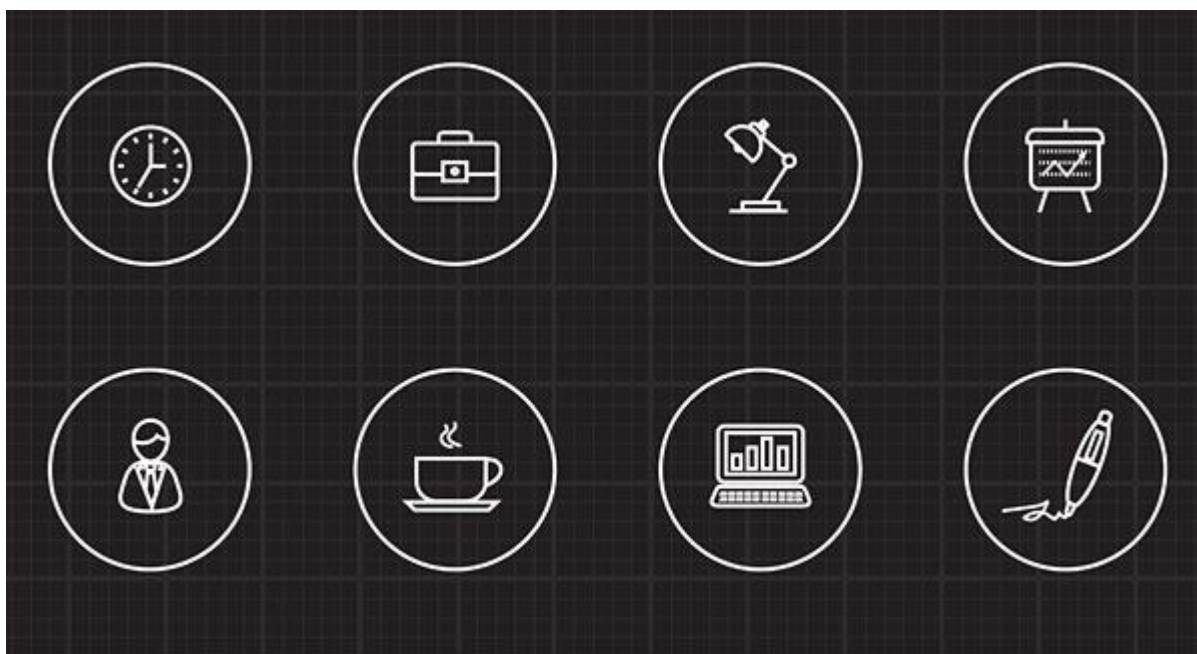


Рис 39. Простые и понятные контурные иллюстрации

– в представлении контента:

Сторителлинг, сайты-истории. Сайт-история сильнее вступает в коммуникацию с пользователем, нежели «просто сайт», где сухо перечисляют преимущества продукта. Сайт-история – это современная форма промо-сайта, требующая серьезной работы с контентом. Тренд получит развитие в будущем. Интерактивный сторителлинг подразумевает, что посетитель не просто листает страницы сайта, а участвует в создании своеобразной истории, что является дополнительным стимулом к взаимодействию с сайтом. Конечно, подходит данный прием далеко не для каждого сайта, но тем не менее он находит свою аудиторию.

Интерактивная инфографика. Инфографика как тренд мощно заявила о себе в прошлом году. Дальше – больше. Сегодня уже стало возможным использовать на сайте интерактивную типографику. Когда, например, при наведении на нужный год или объект меняется соответствующая информация в визуальном блоке.

Новые тренды, в том числе перечисленные здесь, быстро становятся обычным явлением в веб-дизайне. Часть из них вскоре отмирает, заменяется другими, нередко контрастными к предыдущим.

УПРАЖНЕНИЕ 9. Анализ использования современных тенденций веб-дизайна.

1. Создайте в поисковой системе запрос об основных тенденциях и трендах веб-дизайна в текущем году. Ознакомьтесь с авторскими публикациями, найденными по этой теме.
2. Просматривая веб-сайты разных тематик и категорий, акцентируйте свое внимание на наличии (или отсутствии) в их дизайне современных тенденций, описанных в рассмотренных в п.1 материалах. Сделайте вывод о том, сайты каких компаний (организаций) чаще других следуют модным тенденциям и с чем это связано.

Контрольные вопросы по теме «Основные тенденции и тренды современного веб-дизайна».

1. Что означают термины «тенденция», «тренд»?
2. Назовите современные тенденции веб-дизайна в области технологических решений; в оформлении внешнего вида сайтов.

ГЛАВА 3. ПРАКТИКУМ: ОСНОВЫ HTML И CSS В ВЕБ-ДИЗАЙНЕ

Веб-страница представляет собой электронный документ, содержимое которого включает как непосредственно контент страницы, так и правила отображения этого контента, указывающие программам просмотра (браузерам) форму представления описанной в документе информации. Эти правила составляются с помощью языка гипертекстовой разметки HTML (Hyper Text Markup Language).

Разметка – это код, который сопровождает содержание документа и предоставляет браузеру (или другой обрабатывающей программе) информацию о том, как следует интерпретировать те или иные фрагменты содержания.

Язык HTML предназначен в первую очередь для структурной (логической) разметки, т. е. для обозначения каждого элемента в соответствии с его местом и ролью в структуре документа. Структурная разметка не имеет никакого отношения к внешнему виду документа. Документ может быть отображен на экране компьютера, написан от руки или напечатан на пишущей машинке, но все заголовки в нем останутся заголовками, все перечни – перечнями, а цитаты – цитатами. Структурная роль элементов документа, в отличие от оформления, никак не меняется в зависимости от того, какими средствами просматривают этот документ.

Таким образом, *структурная разметка* – это разметка, обозначающая роль каждого элемента в структуре документа. Структурная разметка говорит о том, чем является тот или иной элемент, а не о том, как его следует или не следует отображать. Грамотная структурная разметка обеспечивает независимость документа от устройства вывода.

Структурная разметка не говорит о том, как конкретно должен быть выделен текст. Но это не мешает разработчику оформить его нужным образом.

Оформительские возможности HTML очень скудны. Для эффектного оформления отдельных элементов и придания комплексу веб-страниц единого узнаваемого стиля, используют технологию CSS (Cascading Style Sheets – каскадные таблицы стилей). Таким образом, структурная разметка и оформление дополняют друг друга.

Итак, на сегодняшний день, язык HTML является основой всех размещенных в Интернете электронных документов. Он выступает в роли фундамента, на

базе которого реализуются прочие сетевые программные технологии, призванные повысить общую привлекательность, эффективность и интерактивность носителей информационных данных. В этой связи, веб-мастеру необходимо знание основ языковых средств создания веб-документов, умение использовать эти знания для создания собственных и для анализа чужих решений.

Далее в главе кратко изложены необходимые веб-дизайнеру основы HTML и CSS, оформленные в виде комплекса практических работ, последовательное выполнение которых обеспечит базовую подготовку специалиста созданию и редактированию веб-сайтов.

3.1. Основные разделы HTML-документа. Описание раздела body

Язык HTML представляет собой набор определенных правил, в соответствии с которыми браузер отображает содержимое электронной страницы. Каждому правилу соответствует свое название (символьное обозначение, код), свойство (или набор свойств, называемых параметрами либо атрибутами) и значение (значения этих атрибутов). Для выделения управляющего кода из текстового массива, его символьное обозначение заключается в угловые скобки – `<...>`.

Подобная конструкция: `<набор текстовых символов>` называется *тегом*. Язык HTML не чувствителен к регистру текстовых символов, поэтому название тега и атрибутов допускается набирать как прописными (`<BODY>`), так и строчными (`<body>`) символами.

Теги бывают одинарными и парными:

- пример непарного тега: `<hr>`;
- пример парного тега: `...`.

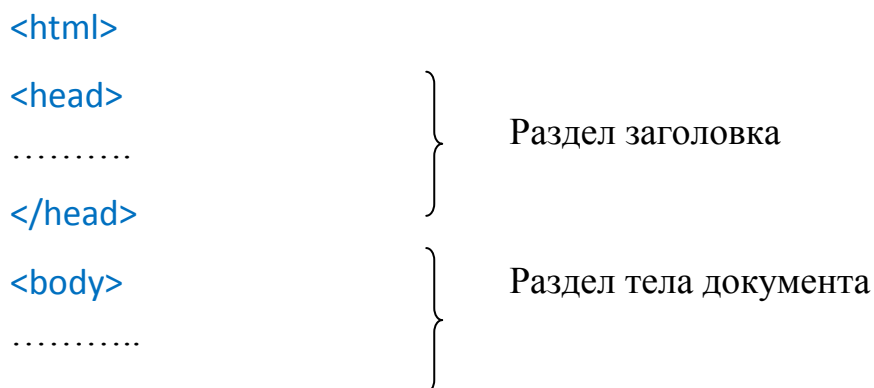
Структура HTML-документа

Документ HTML условно состоит из трех частей:

1. Строка, содержащая декларацию типа документа;
2. Заголовок (шапка) документа;
3. Тело документа.

В первой строке заявляется тип документа (DOCTYPE), а также правила, которым следует разметка, и согласно которым она будет проверяться. Правила оформления строки можно найти на соответствующих ресурсах сети Интернет. Основные правила представлены в Приложении. 1.

Вторая и третья части находятся в границах тега HTML. Таким образом, схема HTML-документа выглядит так:



`</body>`

`</html>`

В *разделе заголовка*, ограниченного контейнером `<head> ... </head>`, могут размещаться определённые теги, которые не вносят изменений в видимое пользователю содержимое веб-страницы, но влияют на связь между документами и оформление гиперссылок. Для оформления заголовка окна документа используется:

`<title> ... </title>` – данный тег-контейнер задает название страницы, которое отображается в строке заголовка окна браузера.

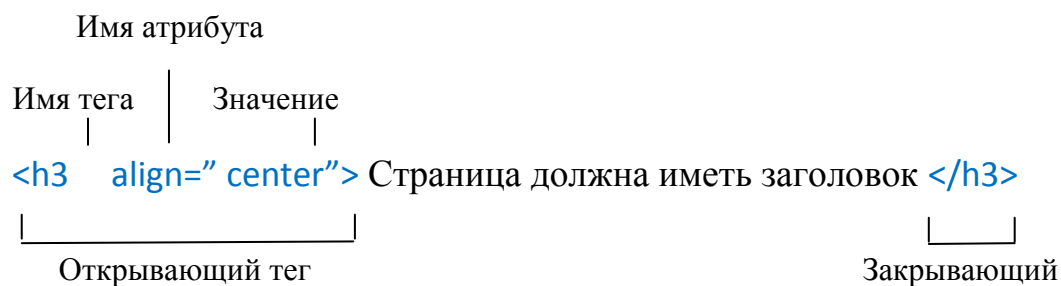
В *разделе тела документа*, ограниченного контейнером `<body> ... </body>` (или `<frameset>...</frameset>` для страниц, созданных на основе фреймов), располагается вся содержательная часть веб-страницы, которая выводится браузером на экран монитора пользователя. Внутри этого раздела может размещаться большинство существующих тегов HTML.

Каждый тег может иметь один или несколько атрибутов.

Атрибуты – имена свойств тегов, которые могут принимать определенные значения.

Атрибуты имеются только у открывающих тегов, у закрывающих тегов их нет. Атрибуты отделяются от тега и других атрибутов пробелами. Если атрибут имеет конкретное значение, то оно задается после знака равенства. Атрибуты влияют на результат интерпретации тега браузером. Регистр написания тегов и атрибутов значения не имеет.

Оформление конструкции тега, содержащего атрибут, выглядит следующим образом:



Описание тела документа.

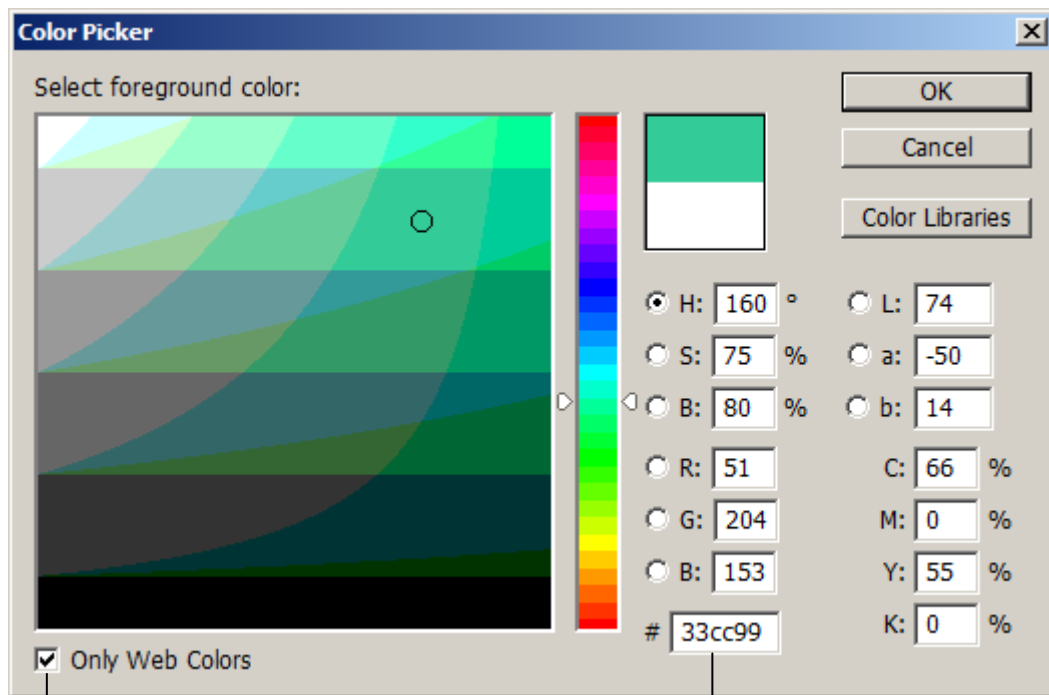
В теге `<body>` задаются свойства документа. Для этого используют атрибуты:

- `bgcolor` – задание цвета заднего фона;
- `background` – задание фонового рисунка;
- `bgproperties` – изменение свойств заднего фона;
- `topmargin`,
- `bottommargin`,
- `leftmargin`, – определение размеров отступов;
- `rightmargin`,
- `marginwidth`,
- `marginheight`
- `text` – задание цвета основного текста;
- `link`, `alink`, `vlink` – задание цвета гиперссылок.

Атрибут `bgcolor` задает однородный цвет фона документа, значение которого может быть введено в символьном виде или в шестнадцатеричном коде:

`<body bgcolor="red">` или `<body bgcolor="#ff0000">`.

При определении цвета элементов HTML-страницы рекомендуется использовать, так называемые, безопасные цвета. Безопасные цвета веб-палитры можно выбрать, например, в окне сборщика цветов программы Adobe Photoshop (рис. 40).



При установленной галочке, в окне выбора цвета отображаются только цвета Web-палитры

Номер выбранного цвета

Рис. 40. Палитра выбора цвета в Adobe Photoshop

При использовании `background`, на задний фон документа помещается графическое изображение, которое, как обои с определенным рапортом, заполнит фон всего документа:

```
<body background="images/logo.gif">
```

Существуют определенные требования к изображениям, помещаемым на HTML-страницы. Файл с изображением должен быть в цветовой модели RGB или Index и сохранен в формате jpeg или gif. Физический размер изображения (размер растровой сетки в пикселах) и размер файла с изображением должны быть приемлемыми для веб-размещения.

`bgproperties` – задает свойство фонового изображения. При значении `fixed` запрещает прокрутку фонового изображения совместно с текстом страницы. Например:

```
<body background="images/logo.gif" bgproperties="fixed">
```

`text` – задает используемый по умолчанию цвет текста страницы, не являющегося гиперссылкой (если атрибут не используется, то цвет по умолчанию – черный). Цвет шрифта может быть задан:

```
<body text="blue"> или <body text="#00ffff">
```

`link`, `alink`, `vlink` – задает цвета: не открытых гиперссылок страницы, активной гиперссылки и посещенной, соответственно. Например, если все не посещенные ссылки требуется отображать синим, активную – красным, а посещенную пурпурным цветами, то можно задать конструкцию:

```
<body link="#0000ff" alink="#ff0000" vlink="#ff00ff">.
```

Атрибуты: `topmargin`, `bottommargin`, `leftmargin`, `rightmargin` – задают: верхний, нижний, левый и правый отступы заданного размера от краев документа. Значения отступов указываются в пикселях:

```
<body topmargin="10" leftmargin="15" rightmargin="15">.
```

Если для описания тела документа необходимо указать несколько атрибутов одновременно, то запись может выглядеть следующим образом:

```
<body background="images/logo.gif" bgproperties="fixed" link="#0000ff" alink="#ff0000" vlink="blue" topmargin="10" leftmargin="15" rightmargin="15">.
```

Дополнительную информацию о структуре HTML-документа и описании его свойств в теге `<body>` можно найти на соответствующих ресурсах сети Интернет, например, на образовательных ресурсах: <http://postroika.ru>; <http://intuit.ru> и пр.

Практикум «Основные разделы HTML-документа. Описание раздела `body`»

1. Создайте в текстовом редакторе (Блокнот) шаблон веб-страницы сохраните его как HTML-документ с именем `listing`. Для этого в окне сохранения документа в поле «Имя файла» введите строчными латинскими буквами без пробелов имя файла и после точки расширение: `html` или `htm`.

Обратите внимание: созданный HTML-документ можно просматривать как с помощью текстового редактора (в виде текста на языке HTML), так и через браузер – как веб-страницу.

2. Создайте веб-страницу, для которой заданы: верхняя, левая и правая границы по 15 пикс., на заднем плане – фоновое изображение, цвет текста – отличный от черного. Сохраните изменения.

3. Откройте сохраненный документ `listing.html` в редакторе Notepad++ (программа предоставляется для свободного использования на официальном сайте <http://notepad-plus-plus.org>)). Обратите внимание на разницу в визуальном представлении текста html-кода. Сделайте вывод об удобстве использования

того или иного текстового редактора для записи HTML-кода гипертекстового документа. В дальнейшей работе используйте выбранный редактор.

4. Скопируйте из любого текстового документа в раздел тела HTML-документа значительное количество текста, которое, при просмотре страницы в браузере, не будет умещаться в одном (первом) экране.

5. Оцените внешний вид полученного документа. Обратите внимание на появление полосы прокрутки и на то, как выглядит документ при прокручивании. Сделайте вывод об удобстве чтения текстовой информации с экрана и о том, как фон и цвет текста влияют на удобочитаемость.

6. Добавьте в код документа атрибут, фиксирующий фоновое изображение. Убедитесь в том, что при прокручивании текста полосой прокрутки, изображение заднего фона остается неподвижным.

7. Откройте созданный HTML-документ в различных браузерах (Internet Explorer, Opera, Safari и др). Сравните вид открытого документа, определите различия, сделайте выводы.

8. Сохраните файл на рабочем столе учебного компьютера для предъявления преподавателю и, при отсутствии замечаний к результатам работы, удалите файл после проверки.

Форма отчета

1. Результатом выполнения работы является содержимое HTML-документа listing.html, которое демонстрируется: 1) в окне текстового редактора; 2) в окне браузера.
2. Содержимое созданного документа должно включать: задание верхней, левой и правой границ документа по 15 пикс. с каждой стороны; установку цвета основного текста, отличного от цвета по умолчанию; установка адекватного (по размерам и цветовому оформлению) изображения на задний фон HTML-страницы, фиксация изображения заднего фона.

Контрольные вопросы

1. Что представляет собой язык HTML? Что называют тегом, атрибутом тега?
2. Какова общая структура HTML-документов? Назовите назначение разделов HTML-документа.
3. Какие свойства документа задаются при описании <body>? Назовите основные атрибуты раздела тела документа и атрибуты тега <body>.
4. Назовите способы задания цвета элементов HTML-документа. Какие цвета называют безопасными?

5. Назовите условия, которым должны удовлетворять графические изображения, предназначенные для размещения на заднем плане (фон) веб-страницы.

Домашняя работа №1

АНАЛИЗ ЗАДНЕГО ФОНА ВЕБ-СТРАНИЦ САЙТОВ РАЗЛИЧНЫХ КАТЕГОРИЙ. СОЗДАНИЕ ФОНОВОГО ИЗОБРАЖЕНИЯ ДЛЯ ВЕБ-СТРАНИЦЫ

Цель работы

Получить представление об особенностях выбора и создания фонового изображения веб-страниц, для чего:

- провести анализ заднего фона веб-страниц сайтов различных категорий;
- на основе результатов проведенного анализа, создать собственное изображение для заднего фона веб-страницы с учетом предполагаемой тематики будущего сайта и различия параметров разрешения мониторов.

Краткие теоретические сведения

Веб-сайт представляет собой совокупность электронных страниц, объединенных по смыслу и стилевому оформлению, связанных системой гиперссылок и физически находящихся на одном сервере. Целостный образ сайта создается сочетанием контента (содержательной части, наполнения) и оформления его страниц.

Для улучшения эстетического вида и узнаваемости веб-страницы, а также для облегчения восприятия информации на ней, используется задний фон (background). Фоном называют графическое изображение, размещенное на заднем плане страницы. Фоновое изображение предназначается для облегчения восприятия информации на странице, а также создания определенного визуального образа. Выбор фонового изображения задает визуальный образ страницы и определяется тематикой (направленностью) ресурса.

В Интернет присутствуют:

- сайты с деловой информацией (сайты биржевых и бизнес-компаний; промышленных, производственных корпораций и др., задачами которых является оперативное и полное представление достоверной информации);

- рекламно-информационные сайты и порталы (сайты производителей и продавцов товаров и услуг массового потребления, правительственные, образовательные, общественные сайты и пр., задачами которых является информирование об определенных сферах общественной жизни);
- развлекательные, художественные сайты (сайты досуговых и развлекательных заведений, электронные выставочные площадки и видеозалы, персональные страницы, являющиеся продуктами самовыражения творческой личности и прочие ресурсы, не несущие серьезной информационной нагрузки, задачами которых является развлечение и неутомительное просвещение посетителя) и др.

Каждой категории сайтов (информационных ресурсов) присущи определенные традиции в оформлении заднего фона страниц, вызванные объективными причинами: задачами сайта, образом компании, предпочтениями целевой аудитории и пр.

Указания к выполнению работы

1. Найдите в сети Интернет страницы веб-сайтов различного назначения (не менее трех примеров сайтов каждой категории). Обратите внимание на оформление фона веб-страниц. Сделайте вывод об особенностях оформления заднего плана страниц для сайтов единой категории.

2. На основе проведенных наблюдений и анализа фоновых изображений рекламно-информационных сайтов, продумайте внешний вид и выполните на бумаге наброски фона воображаемого сайта задуманной тематики.

3. Используя графический редактор Adobe Photoshop, создайте два изображения, предназначенных для помещения на задний фон веб-страницы:

- 1) рисунок фона (фрагмент¹⁰), который будет тиражироваться (клонироваться) слева направо и сверху вниз, заполняя (заливая) все пространство окна браузера;
- 2) рисунок, композиция которого предусматривает корректное покрытие заднего плана веб-страницы без тиражирования (клонирования) при открытии ее (страницы) на экране монитора с размерами от 800×600 до 1920×1080 пикселей.

¹⁰ Приступая к работе, продумайте, как будут совмещаться левая и правая, а также верхняя и нижняя стороны создаваемого рисунка

Форма отчета

1. Результат выполнения пунктов 1 и 2 данной работы представляется в форме электронного текстового документа, содержащего:
 - для каждой группы (категории) сайтов: наименование группы; описание задач, на выполнение которых нацелены сайты этой группы; описание факторов, определяющих предпочтения в оформлении заднего плана веб-страниц; перечень сайтов, рассмотренных в данной группе (с указанием URL-адресов найденных ресурсов); скриншот¹¹ характерного примера страницы сайта данной категории;
 - вывод, сделанный по результатам наблюдений, о замеченных предпочтениях при выборе фона страниц для сайтов каждой категории.
2. Результат выполнения п. 3 Указаний к выполнению работы (далее – Указаний), представляется комплектом файлов, демонстрирующих:
 - образец текстурной (фактурной) заливки веб-страницы (файл формата jpg, gif или png), созданный при выполнении п. 3.1;
 - полноформатный рисунок фона веб-страницы (файл формата jpg, gif или png), созданный при выполнении п. 3.2 Указаний;
 - вид веб-страницы¹², на задний план которой установлен рисунок, созданный при выполнении пунктов 3.1 или 3.2 (файлы формата html).

Контрольные вопросы

1. Назовите особенности визуального оформления заднего плана веб-страниц для сайтов различной направленности.
2. Какие требования предъявляются к рисункам текстуры, автоматически заполняющим задний план веб-страницы? К рисункам, устанавливаемым на задний план страницы без тиражирования?

¹¹ Скриншот (screenshot) - снимок экрана, изображение, полученное компьютером, которое точно отображает то, что показывает монитор в конкретный момент времени.

¹² В раздел <body> демонстрируемых страниц разумно добавить несколько абзацев произвольного текста, чтобы сделать вывод о пригодности фона его читабельности на данном фоне.

3.2. Форматирование текста в HTML

(Логическое и физическое форматирование текста в HTML- документе.

Использование HTML- списков)

В большинстве случаев посетители веб-страниц обращаются на сайт в поисках текстовой информации. Текст является важной составляющей HTML-документа и от того, как он преподнесен читателю, зависит удобство работы с сайтом и в конечном итоге – его успех.

Под форматированием текста понимается возможность с помощью специальных тегов красиво располагать текст на странице, выводить в виде всплывающих подсказок, расставлять смысловые акценты применением различного шрифта, цвета, отступов и т. п.

Структурное форматирование документа

Любой документ, созданный с определенной логикой, имеет свою структуру. Структура учебной книги, например, предполагает деление на главы, параграфы, пункты, которые снабжены заголовками. Кроме того, в ней могут быть выделены практикумы, справочные разделы, вопросы для самоконтроля. Структурное форматирование электронного документа подразумевает разбиение текстовых фрагментов на логические блоки и включает: создание заголовков, установку абзацев, центрирование, отступы, перенос строки, вставку горизонтального разделителя, списков, добавление комментариев и т.п., то есть все то, что облегчает восприятие страницы пользователем и обработку документа автоматическими средствами.

Заголовки

Заголовки – важная составляющая контента веб-страницы: они помогают систематизировать текст. В HTML доступно создание заголовков разных уровней, поэтому легко выделять смысловые разделы и подразделы. При индексации текста страницы поисковыми системами, особое внимание уделяется содержанию заголовков. Текст, оформленный как заголовок, имеет для поисковых роботов более высокий ранг.

В HTML-документе можно создать заголовки шести уровней.

`<h1>`, `<h2>`, ... `<h6>` – используются для обозначения заголовков соответствующего уровня (от 1 до 6). Применение тега `<h>` для оформления заголовка подразумевает отступ от основной текстовой части и задание высоты букв заголовка, соответствующей его уровню (текст заголовка, размещенный в

контейнере `<h1> ... </h1>` будут наибольшими по величине, а в `<h6> ... </h6>` будут отображаться с минимальной высотой букв).

Абзац

В HTML-документе невозможно разбить текст на абзацы используя клавишу Enter. Абзацы задаются использованием тега `<p>`.

Каждому заголовку или абзацу с помощью атрибута `align` может быть задан определенный тип выравнивания.

`align` – атрибут выравнивания. Может принимать значения:

`left` – выравнивание по левому краю (значение по умолчанию);

`center` – выравнивание по центру;

`right` – выравнивание по правому краю;

`justify` – выравнивание по ширине окна браузера.

Обрыв строки

При необходимости оборвать строку не закрывая абзаца, или же для принудительного помещения следующего элемента в начало новой строки, применяется непарный тег `
`.

Центрирование

Центрирование любых элементов HTML-документа может быть осуществлено с помощью тега-контейнера `<center>`. Все данные, размещенные внутри контейнера `<center> ... </center>`, подлежат выравниванию по горизонтальному центру окна браузера.

Отступ

В текстовом редакторе отступ задается клавишей табуляции Tab. В HTML-документе размер дополнительного отступа определяется кодовой конструкцией – символом пробела – ` `. Если необходимо сделать отступ, соответствующий визуально трем пробелам, то для обеспечения смещения потребуется трижды повторить указанный символ: ` `.

Горизонтальный разделитель

Для разделения фрагментов текста в соответствии с логикой подачи материала на странице сайта, веб-дизайнеры проводят визуальную границу между текстовыми блоками, так называемый *горизонтальный разделитель*.

`<hr>` – тег создания горизонтальной разделительной линии.

Тег `hr` может включать ряд атрибутов, описывающих свойства разделительной линии:

`width` – указание длины линии. Значение может задаваться абсолютно (количеством пикселей) или относительно (в процентах от ширины окна браузера).

`size` – указание толщины линии. Значение толщины указывается в пикселях;

`align` – выбор типа выравнивания линии. Типы выравнивания: `left`, `center`, `right`;

`color` – выбор цвета линии;

`noshade` – отмена эффекта рельефности линии.

Конструкция для задания горизонтальной линии зеленого цвета, толщиной 3 пикселя, растянутой на 80% от ширины окна браузера и выровненной по правому краю, будет иметь вид:

```
<hr color="green" size="3" width="80%" align="right">.
```

Предварительно отформатированный текст

Тег-контейнер `<pre> ... </pre>` позволяет выводить на экран монитора текст в таком виде, как он выглядит в окне текстового редактора: со всеми переносами строк, отступами, абзацами и прочим форматированием.

Списки

Списки – важный инструмент, применяемый для организации и группировки данных. В HTML можно выделить списки нескольких типов:

- маркированный – список, пункты которого отмечены с помощью различных символов (применяется когда порядок следования пунктов не имеет принципиального значения);
- нумерованный – список, пункты которого упорядочены и последовательно обозначены цифрами или буквами алфавита (применяемый когда порядок следования пунктов имеет большое значение);
- список определений – вид списка, который применяется для формирования словарей или когда необходимо пояснять значения терминов.

Для задания списков используются теги-контейнеры:

`` – задает маркированный список;

`` – задает нумерованный список;

`<dl>` – задает список определений.

Для тегов `` и `` может задаваться атрибут `type`, определяющий вид маркера или вид нумерации. Значения атрибута `type` для маркированных и нумерованных списков приведено ниже.

| Тип списка | Значение атрибута <code>type</code> | Вид создаваемого маркера |
|----------------|--|---|
| Маркированный: | <code>circle</code> <code>square</code> <code>disc</code> | маркер в виде окружности; маркер в виде квадрата; маркер в виде черного круга. |
| Нумерованный | <code>A</code> <code>a</code> <code>I</code> <code>i</code> <code>1</code> | заглавные буквы латинского алфавита; строчные буквы латинского алфавита; заглавные римские цифры; строчные римские цифры; арабские цифры. |

Для нумерованного списка существует возможность с помощью атрибута `start` указать с какого порядкового номера начать нумерацию всего списка. (Например, при использовании конструкции (`<ol type="a" start="4">`, нумерация будет производиться заглавными строчными буквами и начнется с «D»).

Для задания элемента маркированного или нумерованного списка используют тег `li`. HTML-код оформления списка выглядит следующим образом:

```
<ol type="a" start="4">
```

```
<li> Текст пункта первого;
```

```
<li> Текст пункта второго;
```

```
<li> Текст пункта третьего.
```

```
</ol>
```

Для задания элементов списка определений, вместо тега `li` используют комплект парных тегов:

`<dt>` – используется для задания термина;

`<dd>` – используют для задания определения термина.

HTML-код списка определений выглядит следующим образом:

```
<dl>
```

```
<dt> Термин 1 </dt>
```

```
<dd> Определение термина 1 </dd>
```

```
<dt> Термин 2 </dt>
```

```
<dd> Определение термина 2 </dd>
```

```
</dl>
```

Комментарии

Разработчику веб-страницы, для того чтобы легко ориентироваться в записи HTML-кода документа, может потребоваться добавление в структуру кодовой страницы неотображаемых в окне браузера пояснений. Такие не видимые пользователю при просмотре в браузере пояснения называют комментариями.

Любой комментарий начинается с символов `<!--` и заканчивается `-->`. Между ними находится текст комментария, который может содержать буквы, цифры, символы и прочее, за исключением тегов. В комментарии можно указывать информацию об авторском праве, персональные данные, названия структурных блоков страницы и прочее.

```
<!-- Начало структурного блока -->
```

Код заявленного блока

```
<!-- Конец структурного блока -->
```

Указанные в комментариях данные будут проиндексированы поисковыми системами.

Физическое форматирование текста

К физическому форматированию относится возможность визуально изменять текст, варьируя его параметры (цвет, гарнитуру, размер и т. п.) произвольным образом. Теги физического форматирования предназначены для выделения отдельных текстовых фрагментов различными способами по усмотрению автора документа.

Тег `` – один из основных тегов физического форматирования. С его помощью задаются такие свойства шрифта как гарнитура, цвет, размер. Для этого внутри `font` могут использоваться следующие атрибуты:

face – задает тип шрифта, которым должен отображаться заключенный в контейнер текст веб-страницы. Если на компьютере пользователя не установлен шрифт, указанный в HTML коде, то текст будет показан стандартным (по умолчанию) шрифтом. Допускается указывать сразу несколько значений параметра FACE, чтобы в случае отсутствия на компьютере первого шрифта, браузер отобразил текст вторым шрифтом из списка. Например:

`текст, к которому применяется оформление`

color – задает цвет текста, которым будет написан фрагмент, заключенный в контейнер

size – управляет размером шрифта. Может принимать значения от 1 до 7. Размер шрифта можно указать абсолютным значением (задать число от 1 до 7) или относительным, то есть указать на сколько пунктов больше или меньше текущего должен быть размер шрифта (например, +3 или -2). По умолчанию устанавливается величина шрифта – 3.

Для физического форматирования применяются следующие теги:

`<basefont>` – определение свойств шрифта для всего документа;

`` – задание свойств шрифта (гарнитура, размер, цвет);

``, `` – выделение жирным начертанием;

`<i>`, `` – выделение курсивом;

`<u>` – выделение подчеркиванием;

`<tt>` – имитация телетайпа или моношириного шрифта;

`<big>` – выделение фрагмента текста шрифтом большего размера, чем основной текст;

`<small>` – выделение фрагмента текста шрифтом меньшего размера, чем основной текст;

`` – присвоение фрагменту текста определенных свойств в случае, когда невозможно использование тегов логического форматирования;

`<s>`, `<strike>` – обозначение удаленной информации зачеркиванием текста;

`<sub>` – оформление нижнего индекса;

`<sup>` – оформление верхнего индекса.

Логическое форматирование текста

Логическая разметка предназначена, в первую очередь, для обозначения каждого элемента в соответствии с его местом и ролью в структуре документа. Логическое форматирование предполагает отображение элементов веб-страницы на экране монитора таким образом, как установлено по умолчанию для элементов данного типа (например, заголовков или цитат) в специфике языка HTML.

Логическое форматирование может осуществляться с помощью тегов:

`<abbr>` – определяет текст как аббревиатуру;

`<acronym>` – определяет текст как акроним¹³;

`<code>` – указание фрагмента программного кода;

`` – обозначение удаленной информации;

`` – выделение курсивом;

`` – выделение жирным начертанием.

`<cite>` – выделяет специальным образом (курсивом) цитаты и высказывания, названия библиографических источников

Теги `<abbr>` и `<acronym>` – позволяют создать всплывающие подсказки, с помощью атрибута `title` для указания текста всплывающего пояснения. Результат использования конструкции:

`<acronym title="Hyper Text Markup Language">HTML</acronym>` – один из наиболее популярных и распространённых на сегодня языков разметки текста, используемый для создания web-страниц.

приведен на рисунке 40.

При использовании тега `<abbr>`, поисковые роботы индексируют именно полный вариант расшифровки, определенный атрибутом `title`.

¹³ Акронимы – некие устоявшиеся сокращения, например, СНГ, США

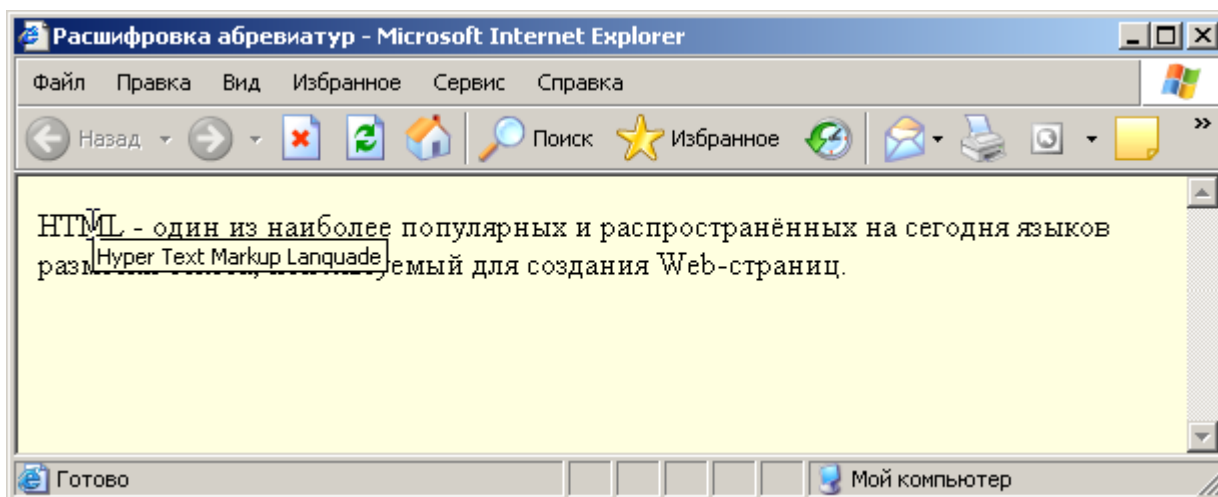


Рис. 40. Результат использования тега <acronym>

Дополнительную информацию тегах структурного, физического и логического форматирования текста можно найти на соответствующих ресурсах сети Интернет.

Практикум «Форматирование текста в HTML»

1. Создайте новый или откройте созданный ранее документ HTML. Опробуйте в действии команды физического, логического и структурного форматирования.
2. Спроектируйте вид веб-страницы, представляющей собой словарь терминов предметной области «веб-дизайн», используя подготовленные текстовые и графический материалы.
3. Применяя команды форматирования, создайте заготовку веб-страницы «Глоссарий¹⁴», наполнение которой будет осуществлено в ходе следующей лабораторной работы (см. приложение 2).

Форма отчета

Результат представляется в виде HTML-файла (или комплекта файлов, если на веб-странице установлено фоновое изображение), содержимое которого демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

Контрольные вопросы

1. Что представляет собой текстовый дизайн сайта?

¹⁴ Глоссарий (лат. glossarium – «собрание глосс») – словарь узкоспециализированных терминов в какой-либо отрасли знаний с толкованием, иногда переводом на другой язык, комментариями и примерами.

2. В чем заключается структурное форматирование текста? Расскажите об использовании тегов <p> и <hr >.
3. В чем заключается физическое и логическое форматирование текста? Расскажите об использовании тегов физического и логического форматирования.
4. Расскажите о правилах оформления маркированных и нумерованных списков.

3.3. Организация гиперссылок

Ценность и популярность веб-сайтов во многом обусловлена наличием гиперсвязей – организованных возможностей быстрого перехода как на внутренние страницы текущего сайта, так и на внешние ресурсы сети. Гиперсвязь обеспечивается формированием гиперссылок.

Особенностью гиперссылки является то, что щелчок мыши по ней приводит к загрузке электронного документа, связанного с исходным при помощи электронного кода. В качестве гиперссылок могут выступать часть текста, изображение или фрагмент изображения, прочие элементы веб-страницы.

На веб-странице можно организовать переходы трех типов:

- *внутренние*, иначе – *якорные* (обеспечивают переход в пределах страницы для документов, размеры которых выходят за пределы одного экрана);
- *локальные* (обеспечивают переход на другие страницы того же самого сайта);
- *внешние*, иначе – *глобальные* (обеспечивают переход на внешние ресурсы сети).

Любая ссылка состоит из указателя ссылки и адреса ресурса, на который необходимо осуществить переход. В качестве указателя ссылки может выступать отдельное слово, фрагмент текста или графическое изображение.

Указатель ссылки описывается парным тегом `<a>`.

Внутри контейнера `<a> ... ` размещают тот текст (рисунок), который на веб-странице должен быть выделен подчеркиванием. При наведении на него курсором указатель будет принимать вид руки с указательным пальцем, свидетельствующей о возможности перехода. Адрес перехода задается с помощью обязательного параметра `href`. Наиболее просто оформляются локальная и глобальная ссылки.

Правила описания локальных и глобальных гиперссылок

Для организации перехода на другую страницу или другой ресурс сети Интернет применяется конструкция:

`Текст, который является ссылкой `

href – указатель адреса перехода по гиперссылке. Значением этого атрибута может быть абсолютный адрес – полный путь к ресурсу или относительный – указание директории размещения требуемого документа относительно текущего.

Пример ссылки с указанием абсолютного адреса (для организации глобальной ссылки):

```
<a href="http://www.omgtu.ru"> Сайт ОмГТУ </a>
```

Пример ссылки с указанием относительного адреса (для организации локальной ссылки):

```
<a href="images/dramteatr.jpg">Омский театр драмы</a>
```

Тег **<a>**, помимо обязательного **href**, может содержать атрибуты:

name – именной идентификатор внутренней ссылки;

title – атрибут создания всплывающей подсказки;

target – атрибут задания окна, в котором браузеру следует открывать указанный в ссылке ресурс. Чтобы не уводить пользователя со страницы, а открывать ссылку в новом окне, параметру задается значение “_blank”.

Правила описания внутренних гиперссылок

Если HTML-документ слишком большого размера и нет возможности разбить его на несколько отдельных файлов, можно предусмотреть переходы, перемещающие пользователя в пределах одной веб-страницы, избавляя его от необходимости прокрутки документа.

Структура внутренней ссылки включает в себя две части:

- саму ссылку,
- именной идентификатор.

Для оформления первой части (самой ссылки), то есть выражения или иного элемента, который будет выделяться как гиперссылка, предусматривается следующая конструкция:

```
<a href="#якорь"> в начало </a>
```

Для обозначения места на странице, на которое нужно перенести пользователя документа со ссылки с указанным в якоря именем, используется конструкция:

```
<a name="якорь"> </a> ... здесь начинается текст, в начало которого был оформлен переход по ссылке.
```

Если в качестве внутренней ссылки будет выступать рисунок (например, изображение стрелочки «вверх» из файла с именем top.gif), то первая часть ссылки будет иметь вид:

```
<a href="#якорь"> </a>
```

Дополнительную информацию об описании гиперссылок можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практикум «Организация гиперссылок»

1. Откройте документ «Глоссарий», созданный в ходе лабораторной работы № 2. В список терминов включите: веб-дизайн; веб-страница; веб-сервис; сайт; и др.
2. Используя ресурсы сети Интернет, отыщите по 2-3 различных толкования каждого термина. Перенесите в созданный HTML-документ найденное определение термина с указанием источника цитирования. Посредством глобальной ссылки, оформите гипертекстовый переход к странице, с которой было осуществлено копирование. Выполните эту операцию для каждого процитированного определения.
3. Оформите якорные ссылки для осуществления быстрых переходов: 1) от термина, стоящего в списке, к участку страницы, содержащему его толкования; 2) снизу вверх, к списку терминов в начале веб-страницы.

Возможный вид страницы глоссария терминов предметной области «Веб-дизайн» представлен в приложении 2.

Форма отчета

Результат представляется в виде папки с файлами, включающей: основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

Контрольные вопросы

1. Что называется гиперссылкой? Что представляют собой внутренние, локальные и глобальные ссылки?
2. Расскажите об организации внутренней и локальной гиперссылок.

3.4. Добавление графики на веб-страницу. Особенности веб-графики

Как любой журнал или проспект теряет привлекательность без цветных иллюстраций, так и HTML-документ кажется невзрачным без использования графики. Под графикой подразумевают значки, рисунки, фотографии и карты-изображения, занимающие часть окна браузера. Графика привлекает внимание посетителя, заостряет его на наиболее важных моментах.

Для вставки графических изображений в HTML-документ используется специальный тег ``, который не требует наличия закрывающего тега.

На сегодняшний день для создания графических изображений, предназначенных для размещения в электронных документах, используются три основных формата: GIF, JPG и PNG.

Тег `` имеет ряд атрибутов, которые задают оформление добавленного на веб-страницу изображения:

`src` – обязательный атрибут указывающий путь (относительный или абсолютный) к файлу изображения.

Если путь к файлу задан простым указанием имени файла:

```
,
```

то при подобном задании значения `src` браузер станет искать файл `logo.gif` в том же каталоге, что и HTML-документ, в котором данный рисунок будет расположен. Обычно для графических изображений выделяется специальная папка – `images`, и тогда путь к файлу прописывается так:

```

```

`border` – задает толщину рамки, отображаемой вокруг рисунка.

Толщина указывается в пикселях, по умолчанию (если параметр пропущен) рамка не отображается. Если графическое изображение является гиперссылкой, то браузер автоматически отобразит вокруг него рамку определенного цвета (обычно-синего) толщиной в 1 пиксел. Если необходимости в рамке вокруг графического указателя ссылки нет, следует установить значение параметра `border` равное нулю.

`width`, `height` – определяют ширину и высоту встраиваемого графического изображения.

Значения ширины и высоты выводимого рисунка, как правило задаются в пикселях. Если данные параметры не указаны, браузером используются оригинальные размеры рисунка. Если указано значение только одного из

параметров, браузер самостоятельно изменит второй параметр, сохраняя пропорцию. Если указаны значения обоих параметров, браузер отобразит рисунок заданных размеров, даже если изображение при этом сильно исказится

align – определяет выравнивание рисунка относительно текста и прочих элементов документа. Возможны следующие значения атрибута:

left – текст обтекает рисунок по правому краю (изображение слева);

right – текст обтекает рисунок по левому краю (изображение справа);

top – верхний край рисунка выравнивается по самому высокому элементу строки;

texttop – верхний край рисунка выравнивается по самому высокому текстовому элементу строки;

absmiddle – выравнивание середины рисунка по середине строки;

bottom – выравнивание нижнего края рисунка по базовой линии строки;

hspace, **vspace** – определяют размер горизонтального и вертикального отступов от встраиваемого изображения до текста. Значения атрибутов задаются в пикселах. По умолчанию значения равны нулю.

alt – необязательный параметр, задает альтернативный (поясняющий) текст. Обычно это короткое описание изображения, которое появляется на месте рисунка в то время, пока идет загрузка изображения или если у пользователя отключена функция демонстрации графики.

title – отображает текст подсказки к рисункам при наведении курсора. Работает только при загруженном изображении.

Конструкция тега вставки рисунка в HTML-документ может иметь вид:

```

```

В соответствии с указанными параметрами, на странице появится изображение из файла **img1.jpg**, находящегося в папке **images**. Рисунок будет занимать площадку 100×150 пикселей, у левого края окна браузера и обтекаться текстом с отступами по 15 пикселей сверху, снизу и справа. Надо заметить, что в данном случае, заданный параметром **hspace** горизонтальный отступ, отодвинет рисунок на 15 пикселей от левого края документа.

При добавлении графики на веб-страницу следует учитывать несколько факторов:

- размер файла изображения;
- разрешение рисунка (размер растровой сетки изображения);
- формат файла изображения;
- цветовую модель, в которой сохранено изображение.

О некоторых из них было сказано выше (см. Л.р №1), остальные рассмотрим подробнее.

Размер файла изображения

Передача большого файла картинки при низкой пропускной способности канала ощутимо уменьшает скорость загрузки страницы и, кроме того, неоправданно увеличивает трафик. Так, например, если оригинальное изображение из файла [img1.jpg](#) имеет разрешение 300×450 пикселей, при этом размер файла составляет 7,24кБ, то тоже изображение с той же компрессией в требуемых габаритах 100×150 пикселей, может иметь размер файла 1,87 кБ, то есть почти в 3,8 раз меньше.

Если на странице использовано несколько изображений, то размер файла каждого из них будет включен в суммарный размер загружаемой страницы, что, безусловно, отразится на трафике¹⁵. В этой связи уместно заранее уменьшать размер изображения в файле до значений, необходимых для верстки.

Для уменьшения размеров картинки можно воспользоваться растровым редактором Adobe Photoshop (рис. 41). В диалоговом окне Image Size в полях Width и Height нужно ввести требуемые значения ширины и высоты в пикселях. Изменяя размеры изображения важно учитывать сохранение первоначальных пропорций изображения.

¹⁵ Трафик – количество данных, которые проходят через сервер за определенное время. Измеряться он может в байтах, килобайтах, мегабайтах и т.д.

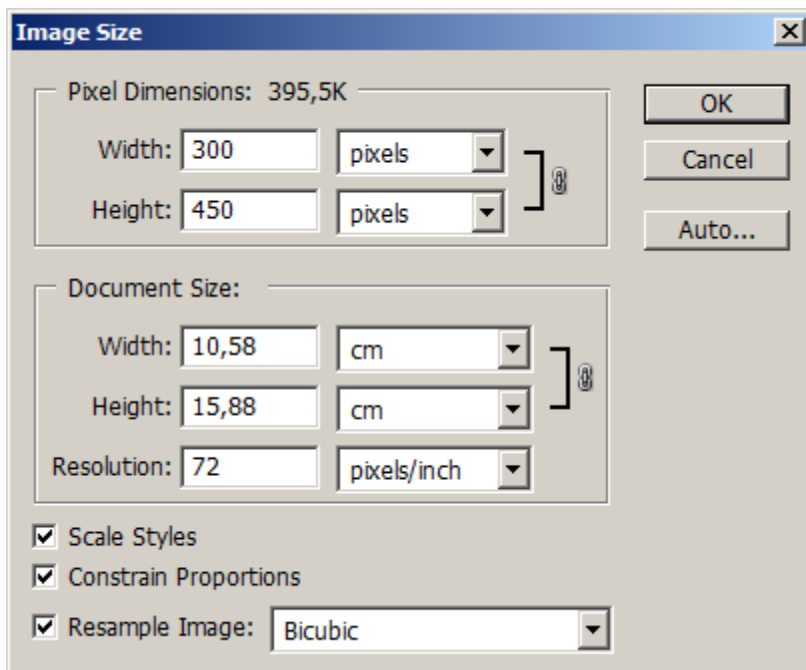


Рис. 41. Диалоговое окно редактора Adobe Photoshop, предназначенное для изменения размеров изображения

Указание габаритов рисунка

Если размеры изображения предварительно изменены до требуемых, то параметры `width` и `height` можно не указывать. Однако использование этих параметров все же рекомендуется, поскольку при указании в теге `` значений ширины и высоты, во время загрузки изображения браузер сразу резервирует на странице столько места, сколько необходимо для отображения рисунка. В противном случае, при загрузке под рисунок резервируется слишком мало места, а потом когда браузер подставляет под маленькое пространство реальные размеры картинка, страница начинает «скакать».

Сохранение пропорций

Размещая на странице уменьшенный вариант изображения, используя указание требуемых габаритов атрибутами `width` и `height`, следует проследить за сохранением пропорций. Так, например, если размер встраиваемого изображения составляет 300×450 пикселей, а конструкции тега указано `width="100" height="180"`, то отображенная браузером картинка будет заметно искажена – растянута по вертикали.

Чтобы уменьшенный рисунок отображался корректно и случайные искажения были предупреждены, можно указать в теге `` только один из параметров (`width` и `height`), значение другого браузер изменит самостоятельно, сохраняя пропорции.

Задание относительных размеров изображения

В случаях, когда браузеру следует изменять размеры рисунка в соответствии с размерами открытого окна, можно задавать значения параметров **width** или **height** в процентах. Такой подход не рекомендован, поскольку некоторые браузеры не поддерживают такое обозначение размеров, либо выводят их некорректно).

Форматы файлов изображений

На сегодняшний день для создания графических изображений, предназначенных для размещения на веб-страницах, используются три основных формата: jpg, gif, png, (wbmp, ico).

JPEG (Joint Photographic Experts Group)

Формат JPEG (для публикации в сети используют расширение *.jpg*) был разработан для передачи фотографий между различными платформами. Благодаря возможностям хранить изображения с 24-битовой глубиной цвета и обеспечивать высокую степень компрессии данных, формат JPEG получил широкое распространение для отображения в сети иллюстраций фотореалистичного качества. Применяемый в JPEG алгоритм сжатия (собственно, JPEG – это как раз название метода сжатия) позволяет варьировать степень компрессии файла в зависимости от требуемого качества отображения иллюстрации.

Кроме того, формат JPEG поддерживает опцию *прогрессивный JPEG*, которая позволяет показывать картинку за несколько проходов до того, как она полностью загрузится на клиентский компьютер. В файл JPEG записывается несколько иллюстраций полного размера, но различного уровня качества. Сначала отображается рисунок низкого качества (и, соответственно, малого объема), а затем за несколько итераций (обычно догружаются 3–5 все более качественных изображений), он превращается в обычный (рис. 42). Это актуально при использовании файлов большого размера и низкой скорости связи, поскольку позволяет увидеть содержание изображения еще до его полной загрузки. При этом общий объем загружаемого файла возрастает незначительно по сравнению с оригинальным JPEG.



а

б

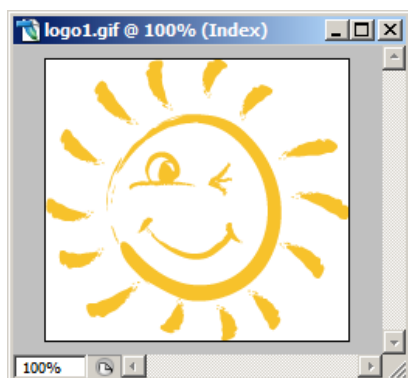
Рис. 42. Изображение фотографии на начальном этапе (а) и после окончания загрузки файла (б)

GIF (Graphics Interchange Format)

Формат GIF (*расширение .gif*) был разработан компанией CompuServe Inc для передачи графической информации с использованием компьютерных сетей. В основе GIF лежит подход, основанный на том, что не каждое цветное изображение должно являться полноцветным. Это значит, что если количество цветов, участвующих в формировании растрового изображения, без существенной потери качества, может быть ограничено до 256, то можно изменить способ кодирования цвета такого изображения и, тем самым, существенно уменьшить размер графического файла. Таким образом, GIF поддерживает до 256 произвольных цветов, что делает его не слишком пригодным для передачи фотографических изображений, но является вполне достаточным для создания графических меню, фоновых текстур, различных рисованных элементов оформления веб-страницы.

Формат GIF обладает рядом свойств, определивших его широкую популярность в веб-дизайне:

Прозрачность. Компьютерные изображения по своей природе всегда прямоугольны, но иногда требуется скрыть это, и тогда фон изображения делают совпадающим с фоном веб-страницы (рис. 43.1). Изображения в формате GIF могут иметь прозрачные области, что позволяет отойти от стандартных прямоугольных форм и создавать элементы страницы произвольной формы которые могут размещаться на многоцветном фоне (рис. 43.2).

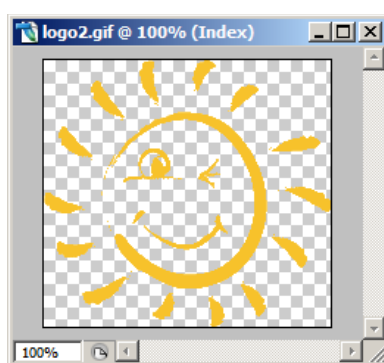


а



б

Рис. 43.1. Белый цвет фона рисунка (а) совпадает с цветом фона страницы, поэтому прямоугольные границы картинки не видны (б)



а



б

Рис. 43.2. Прозрачный фон (а) скрывает прямоугольную форму рисунка, поэтому истинные границы картинки не читаются на цветном фоне веб-страницы (б)

Чередование строк. Графический файл может быть записан таким образом, что при отображении сначала будет загружаться каждая четвертая строка, растянутая на три соседних (изображение низкого качества). Затем, по мере загрузки последующих строк, изображение примет окончательный вид. Такая схема позволяет пользователю получить представление об изображении еще до окончательной загрузки графического файла.

Анимация. Форматом GIF предусмотрена возможность создания анимированных изображений. Для этого это несколько слоев изображения, записанных в один файл, чередуются в определенной последовательности с заданным интервалом времени.

Компактность. Возможность ограничения цветовой палитры только цветами, участвующими в изображении или преднамеренное уменьшение количества цветов до 2, 4, 8, 16, 32 и т.д., уменьшает объем сохраняемых данных. Наличие специального алгоритма сжатия данных позволяет подвергать компрессии файлы без изменения качества.

PNG (Portable Network Graphics)

Формат PNG (*расширение .png*) разрабатывался для передачи графики по сети, с учетом всех особенностей Интернета. Он вобрал в себя преимущества предыдущих стандартов и исключил некоторые их недостатки.

Формат PNG поддерживает изображения как с полноцветной палитрой (PNG-24), так и с пользовательской палитрой цветов (PNG-8).

PNG поддерживает многоуровневую прозрачность, что позволяет создавать плавный переход от прозрачной области изображения к цветной (градиент).

Реализованный в PNG алгоритм сжатия позволяет сжимать рисованные изображения без потери качества лучше, чем GIF (от 5 до 25%).

В PNG применяется механизм чересстрочного вывода первой картинке с использованием двумерной чересстрочности, что позволяет обеспечить минимальные затраты времени на вывод первого изображения.

Кроме того, в PNG реализован эффективный механизм контроля целостности файла, позволяющего обнаружить ошибки при передаче по сети.

Форматом PNG не поддерживается анимация.

Несмотря на некоторые преимущества данного формата, он не получил такого широкого распространения в Интернет, как форматы GIF и JPEG. Однако, этот формат активно используется в специфических направлениях интернет-отрасли (например, в работе программного модуля GD::Graph, автоматически генерирующего графики и диаграммы для систем статистики, рейтинга и пр.)

Дополнительную информацию о добавлении изображений на веб-страницу можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практикум «Добавление графики на веб-страницу. Особенности веб-графики»

1. Откройте созданный ранее документ– Глоссарий терминов. Добавьте к текстовому содержимому веб-страницы иллюстрации по смыслу.
2. Поэкспериментируйте с размерами и размещением иллюстраций, варьируя значения атрибутов тега `` и место вставки тега в HTML-документ.
3. Придайте веб-странице законченный вид, гармонично совместив иллюстрации и текст.
4. Добавьте к внутренним ссылкам, обеспечивающим быстрое перемещение вверх страницы, графические элементы – стрелки.

Форма отчета

Результатом выполнения работы является папка с файлами, включающая: основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

Контрольные вопросы

1. Опишите способ добавления графики на веб-страницу тегом `img`. Расскажите об особенности совместного использования атрибутов `width` и `height` тега `img`.
2. Какими характеристиками должен обладать графический документ, размещаемый на веб-странице?
3. Назовите форматы файлов графических данных, пригодные для размещения на веб-странице, область их применения.

3.5. Метаданные веб-документов

Внутри раздела `head` можно размещать теги HTML: `<title>`, `<meta>`, `<link>`.

Тег TITLE

Парный тег `<title> ... </title>` используется для указания имени созданного документа. Он отображается как заголовок страницы или название окна документа.

`<title>intuit.ru: Учебный курс - Спецификация языка HTML</title>`

Под именем документа в данном случае имеется в виду не имя файла, а визуальный заголовок HTML-страницы (рис. 43).

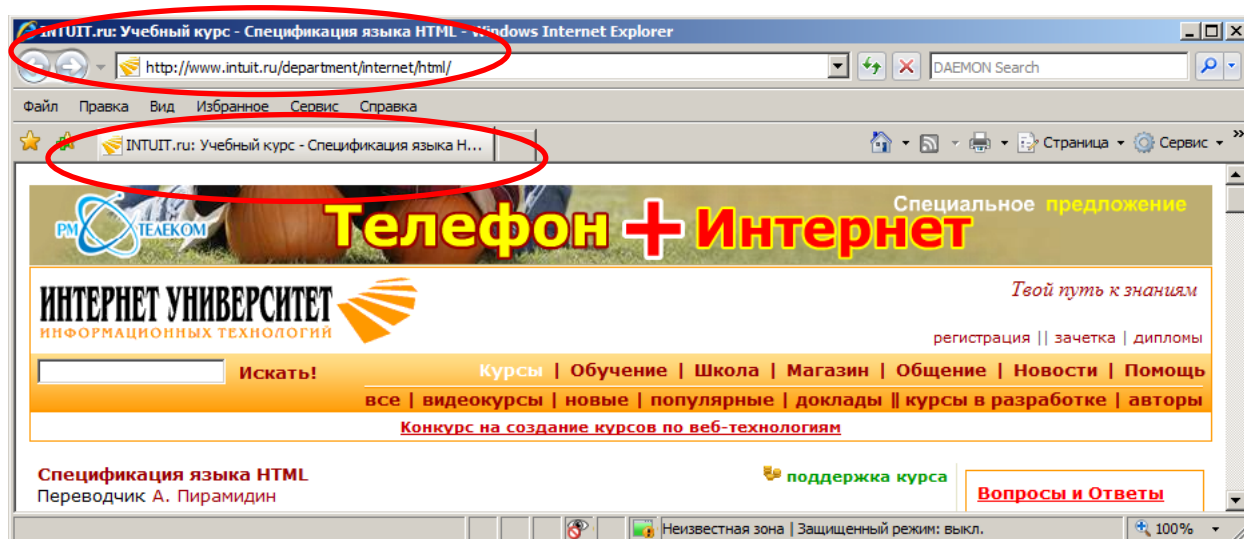


Рис. 43. Содержимого тега `<title>` отображено в заголовке окна браузера и названии текущей вкладки

Тег `<title>` в HTML-документе можно использовать только один раз. Указание в заголовке конструкции `<title> ... </title>` не является обязательным, однако рекомендуется по ряду причин:

- при отсутствии тега названия документа, браузер при интерпретации HTML-кода выведет в заголовке окна фразу типа «Untitled Document» («Документ без названия»);
- при попытке добавить созданную без тегов `<title> ... </title>` страницу в «закладки» браузера, пользователю придется самостоятельно вписывать название добавляемой страницы;
- поисковые системы, индексируя безымянный документ, занесут его в базу под заголовком `Untitled`, что сделает документ безликим и затруднит его поиск через поисковые сервисы.

Метаданные

Метаданными называется информация о внутренних свойствах документа, а не о его содержимом. Спецификация метаданных состоит из двух шагов:

4. Объявление свойства и значения этого свойства;

5. Ссылка на профиль, где определены свойства и их действительные значения.

Объявить свойства и их значения можно двояко:

- задать их в текущем документе через элемент `meta`;
- объявить их во внешнем документе и связать метаданные через элемент `link`.

Метаопределения META

Метаданные могут вводиться в HTML-документ при помощи непарного тега `<meta>`. Все метаопределения имеют два основных типа данных: `name` и `http-equiv`. Атрибуты `http-equiv` и `name` являются двумя возможными параметрами тега `<meta>`, но поскольку они имеют разное предназначение, указывать их одновременно в рамках одного тега нельзя. `<meta http-equiv ...>` переопределяет переданные сервером http-заголовки, `<meta name ...>` устанавливает метаданные документа.

Структура метаданных типа – `name`. Конструкция метатега выглядит так:

```
<meta name="имя" content="содержание">
```

Конструкции этого типа, могут задавать:

- набор ключевых слов, предназначенных для индексирования документа поисковыми системами. Слова указываются через запятую:

```
<meta name="keywords" content="учебный курс, спецификация языка, html, intuit">
```

- краткое описание данной веб-страницы, которое будет отображаться поисковыми системами на странице выдачи результатов рядом со ссылкой на найденный интернет-ресурс:

```
<meta name="description" content="Обучающий курс построен по спецификации, которая определяет HTML 4.01 HyperText Markup Language (HTML) - гипертекстовый язык разметки, язык World Wide Web">
```

- указание автора (авторов) текущего документа:

```
<meta name="Author" content="Иванов, Петров, Сидоров">
```

- описание правил индексирования документа поисковыми роботами:

`<meta name="robots" content=" значение">`

Возможные значения параметра CONTENT: [All](#), [None](#), [Index](#), [Noindex](#), и др.

Если конструкция `meta name="Robots"` не указывается, то поисковый робот индексирует документ со всеми присутствующими в нём гиперссылками, считая принятым по умолчанию значение [All](#) или [Index, Follow](#).

Теперь рассмотрим метаопределения [http-equiv](#). Конструкция метаданных выглядит аналогично предыдущей:

`<meta http-equiv="имя" content="содержание">`

Значениями атрибутов HTTP-EQUIV можно задавать:

[Expires](#) – запрет кеширования HTML-страницы по истечении указанного срока

[Refresh](#) – перезагрузка/переадресация через заданный промежуток времени;

[Content-Type](#) – определение типа и кодировки документа;

[Content-Language](#) – указание языка документа;

[Canche-Control](#) – контроль кеширования документа;

[Window-target](#) – определение места загрузки документа (для фреймовых структур);

[Pragma](#) – управление кешированием по протоколу HTTP/1.0;

[Set-Cookie](#) – настройка чтения данных Cookies пользовательского компьютера;

[Ext-cache](#) – управление альтернативным кешем доукмента;

[Location](#) – указание полного адреса расположения документа в сети Интернет.

Рассмотрим применение некоторых конструкций.

`<meta http-equiv=" expires" content="Mon, 28 Dec 12:00:00 GMT">`

Данная конструкция указывает дату устаревания документа и запрещает кеширование страницы после указанного времени. При повторном обращении к документу по истечении заданного в параметре `content` срока, браузер будет перенаправлен к источнику для обновления информации. Значение параметра `content` содержит следующую последовательность: день недели (Mon, Tue и т. д.), число (01, 02, 03 и т. д.), месяц (Jan, Feb, и т. д.), год, время (чч:мм:сс) и часовой пояс (GMT).

`<meta http-equiv="refresh" content="10; URL=http://www.omgtu.ru">`

Такая конструкция предписывает браузеру перезагружать содержимое окна через заданный промежуток времени (в секундах). Если дополнительно

добавлен подпараметр `URL`, то через указанное время будет произведена переадресация на указанный внешний адрес.

```
<meta http-equiv=" content-type" content="text/html; charset=Windows-1251">
```

Эта конструкция определяет тип и кодировку документа. Наиболее используемыми значениями кодировки русскоязычных документов являются `Windows-1251` и `KOI8-R`. Кодировка указывается посредством подпараметра `charset`. На некоторых серверах возможно автоматическое перекодирование документа, в связи с чем, использовать эту конструкцию не рекомендуется. Важно чтобы кодировка текста документа совпадала с кодировкой, обозначенной в значении `charset`.

Дополнительную информацию об описании элемента `META` можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Связь между документами LINK

Если веб-страница имеет структурные или функциональные связи с другими документами, то для описания этих связей используется не парный тег `<link>`. В конструкции этого тега могут использоваться следующие атрибуты:

`href` – указание ссылки на внешний документ;

`type` – тип внешнего документа;

`rel` – отношение между текущим и внешним документом;

`rev` – отношение между внешним и текущим документом.

Указать на связь текущей веб-страницы с внешним файлом шаблона стилей `main.css` можно, поместив в разделе `head` конструкцию:

```
<link href="styles/main.css" type="text/css" rel="stylesheet">
```

Дополнительную информацию об описании элемента `LINK` можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практикум «Метаданные веб-документов»

1. Для выполнения работы необходимо выбрать веб-страницу рекламно-информационного сайта, размещенного в сети Интернет. Для этого, используя любую поисковую систему, введите ключевые слова запроса и осуществите поиск ресурса по выбранной теме. Зафиксируйте для предстоящего анализа следующие сведения: введенные в строку поиска ключевые слова; заглавие и текст описания страницы, которая показалась наиболее соответствующей

поисковому интересу. Сделайте и сохраните скриншот страницы поисковой системы, содержащей результат выдачи.

2. Перейдите по ссылке на выбранную веб-страницу. Сравните текст заголовка открытой веб-страницы с текстом заголовка, размещенного в листе выдачи поисковой системы и с текстом введенного поискового запроса. Сделайте и сохраните скриншот веб-страницы.

3. Откройте исходный код выбранной веб-страницы. Сравните текст заголовка с содержимым тега-контейнера `<title>`. Сравните текст описания страницы в результатах выдачи с содержимым МЕТА-тега, включающего атрибут `name="description"`, а также с текстом введенного поискового запроса. По результатам сравнения сделайте вывод.

4. Рассмотрите содержимое МЕТА-тега, включающего атрибут `name="keywords"` в исходном коде страницы. Обратите внимание, совпадают ли введенные в поисковый запрос слова с заявленными в значении атрибута `content`. По результатам сравнения сделайте вывод.

5. Рассмотрите содержимое остальных МЕТА-тегов данной веб-страницы. Рашифруйте все метаопределения, заданные для этой страницы. Опишите полученные результаты в тексте отчета. Скопируйте фрагмент исходного кода страницы, включающий описание раздела заголовка (`<head>`).

6. Создайте документ с описанием хода и результатами выполнения лабораторной работы. Сформулируйте и запишите вывод.

Форма отчета

Результат выполнения работы представляется в форме распечатанного текстового документа, содержащего следующие элементы:

- скриншот страницы выдачи поисковой системы, содержащий поле с текстом введенного запроса и блок с описанием страницы, выбранной для анализа;
- скриншот открытой страницы, выбранной для анализа;
- листинг фрагмента HTML-кода страницы (раздел заголовка), выбранной для анализа.

Контрольные вопросы

1. Что называют метаданными документа? Назовите теги описания метаданных HTML-документа.
2. Какая информация об HTML-документе описывается тегом МЕТА? Опишите конструкцию тега МЕТА.

3. Какую информацию, содержащуюся в HTML-коде страницы, используют поисковые системы при обработке поискового запроса?

3.6. Организация таблиц. Использование табличных структур для верстки веб-страницы

В HTML таблицы являются удобным средством структурирования и представления информации и составляют основу большинства HTML-документов.

Использование таблиц позволяет:

- точно размещать отдельные элементы страницы относительно друг друга и границ документа;
- размещать разнородные блоки текста на одном уровне в виде отдельных колонок;
- сохранять дизайн веб-страницы неизменным в ситуации непредсказуемого изменения размеров окна браузера пользователем и пр.

Именно поэтому большинство HTML-документов создается на основе таблиц, где в качестве несущей основы берется таблица с невидимыми краями, содержащая вложенные таблицы с разным оформлением и отличающимися значениями параметров.

Создание простейших таблиц

Для создания таблицы используется тег-контейнер `<table> ... </table>`.

Таблицы состоят из рядов строк, содержащих произвольное количество ячеек. Описание каждого нового ряда осуществляется внутри тега-контейнера `<tr>...</tr>`. Описание ячеек таблицы осуществляется посредством тегов-контейнеров `<td>` и `<th>`. При этом тегом `<td>` описываются ячейки с данными (текстовыми или графическими), а тегом `<th>` – ячейки с заголовками.

Текст, размещенный в `<th>`, будет по умолчанию отображаться жирным начертанием с типом выравнивания «по центру», а данные в `<td>` – обычным начертанием с выравниванием по левому краю.

Таблице можно присвоить заголовок, записав его внутри контейнера `<caption> ... </caption>`.

Простейшая таблица, состоящая из одной строки, содержащей две ячейки задается следующим образом:

```
<table>
```

```
<caption>Пример простейшей таблицы</caption>
```

```
<tr>
<th>Ячейка, содержащая заголовок </th>
<td>Ячейка, содержащая данные</td>
</tr>
</ table>
```

Тег `<table>` является основным и может включать в себя ряд атрибутов:

`border` – задает рамку, показывающую внутреннюю структуру таблицы. Значение толщины бордюра указывается в пикселах. Если атрибут не задан, то рамка не будет отображаться браузером. При использовании атрибута без указания значения параметру `border` присваивается значение `border="1"`;

`bordercolor` – определяет цвет рамки вокруг таблицы. Значения указываются в шестнадцатиричном коде (например, `"#6600ff"`) или в виде наименования цвета (`"green"`);

`cellspacing` – задает расстояние между соседними ячейками таблицы по вертикали и горизонтали, значение указывается в пикселах;

`cellpadding` – задает отступ между рамкой ячейки и её содержимым по вертикали и горизонтали. Значение указывается в пикселах. Если установить `cellpadding="0"`, то содержимое будет вплотную прилегать к рамке.

`align` – устанавливает выравнивание всей таблицы относительно ширины документа. Может принимать значения:

`left` – выравнивание по левому краю документа (значение по умолчанию),

`right` – выравнивание по правому краю документа,

`center` – центрирование таблицы относительно ширины документа.

`width`, `height` – параметры ширины и высоты таблицы. Значения этих параметров можно задавать в абсолютных (пикселах) или относительных величинах (процентах от ширины окна браузера). Например:

```
<table width="600" height="400">
```

```
<table width="80%">
```

При использовании процентов для указания размеров таблицы размеры ячеек будут изменяться соответственно изменениям размеров окна браузера.

bgcolor, **background** – используются для установления заднего фона всей таблицы, отдельных ячеек или отдельных строк.

Внутри тегов **<tr>**, **<td>** и **<th>** также могут располагаться некоторые атрибуты, определяющие вид содержимого ячеек:

bgcolor – задает цвет заднего фона. Используется для тегов **<tr>**, **<td>** и **<th>**.

background – помещает на задний план указанное изображение. Используется только для тегов **<td>** и **<th>**.

align – указывает тип выравнивания содержимого ячейки по горизонтали, принимает значения: **“left”**, **“right”**, **“center”**.

valign – указывает тип выравнивания содержимого ячейки по вертикали, может принимать значения:

top – выравнивание по верхнему краю ячейки,

middle – выравнивание по середине ячейки (значение по умолчанию),

bottom – выравнивание по нижнему краю ячейки,

baseline – выравнивание по базовой линии, обеспечивающее привязку текста одной строки к единому уровню.

width, **height** – параметры ширины и высоты отдельных ячеек или строк.

nowrap – запрет переноса строки в ячейке или табличном ряду.

Объединение ячеек таблицы

В тегах **<td>** и **<th>** возможно использование атрибутов объединения ячеек таблицы:

colspan – показывает, сколько ячеек по горизонтали должны объединиться в одну;

rowspan – показывает, сколько ячеек по вертикали должны объединиться в одну.

Прорисовка структуры таблицы

Внутри тега **<table>** можно использовать атрибуты, отвечающие за внешний вид линеек таблицы.

frame – параметр прорисовки рамки вокруг таблицы. Может принимать значения:

box, **border** – рамка с четырех сторон,

above – линия рамки только сверху,
below – линия рамки только снизу,
hsides – только верхняя и нижняя линии рамки (горизонтальные),
vsides – только правая и левая линии рамки (вертикальные),
lhs – линия рамки только слева,
rhs – линия рамки только справа,
void – нет рамки.

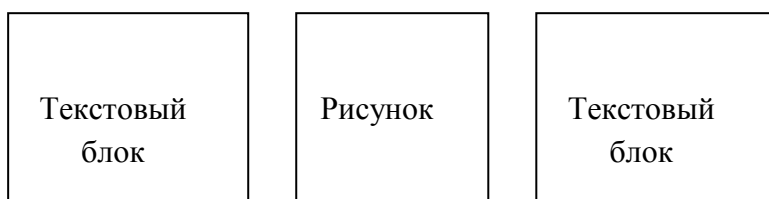
rules – параметр прорисовки внутренних линеек таблицы. Может принимать значения:

all – отображение линеек целиком,
cols – отображение только линеек, разделяющих столбцы,
rows – отображение только линеек, разделяющих строки,
none – отсутствие внутренних линеек.

Дополнительную информацию об описании гиперссылок можно найти на соответствующих ресурсах сети Интернет, а также в учебной литературе.

Практикум «Организация таблиц. Использование табличных структур для верстки веб-страницы»

1. Создайте или откройте HTML-документ. Разместите внутри документа данные с использованием табличной структуры, таким образом, чтобы это соответствовало схеме:



2. Поэкспериментируйте с размерами, размещением и внешним видом таблицы, варьируя значения атрибутов тегов `<TABLE>`, `<TR>`, `<TD>` и `<TH>`.

3. Для страницы «Глоссарий терминов» создайте несущую основу – таблицу с невидимыми краями, ограниченную по ширине, указанием абсолютного значения ширины таблицы около 800 пикселей.

Форма отчета

Результатом представляется в виде папки с файлами, включающей: основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

Контрольные вопросы:

1. Назовите преимущества использования таблиц для верстки HTML-документа.
2. Какие атрибуты возможно использовать одновременно в тегах `<table>`, `<tr>`, `<td>` и `<th>`? Чем будет отличаться результат применения атрибута в этих случаях?
3. Каким приемом можно сделать видимой табличную структуру страницы веб-сайта?

3.7. Вложенные таблицы. Табличная верстка веб-страниц

В HTML одна таблица может включать внутри себя другую, а та, в свою очередь, еще одну, создавая таким образом табличные структуры многоуровневой вложенности. Использование вложенных таблиц позволяет просто и точно размещать отдельные элементы страницы относительно друг друга и границ самого документа, отображаемого браузером. Можно констатировать, что значительное число HTML-документов создается на основе таблиц, где в качестве несущей основы берется таблица с невидимыми краями, содержащая вложенные таблицы с разным оформлением и отличающимися значениями параметров.

Правила построения вложенных таблиц ничем не отличаются от создания таблиц одного уровня: используются те же теги и параметры, задаются те же свойства и значения. Каждая вложенная таблица размещается внутри тега-контейнера `<td>` или `<th>` таблицы предыдущего уровня. Пример организации многоуровневой таблицы, приведенной в приложении 3:

```
<html>
<head>
<title>Использование многоуровневых вложенных таблиц</title>
</head>
<body bgcolor="#ccccaa" text="black" link="#00ff00" alink="00ff00" vlink="blue" >
<h2 align="center"> ПОСТРОЕНИЕ ВЛОЖЕННЫХ ТАБЛИЦ</h2>
<!--Основная несущая таблица-->
<table align="center" border="0" cellspacing="0" cellpadding="2" width="100%" >
<tr>
<td valign="top" width="50%">
<!--Левая таблица с текстом. Начало-->
<table align="center" border="1" cellspacing="3" cellpadding="5" width="100%" >
<tr>
<th bgcolor="gray"><font color="white">Вложенные таблицы</font></th>
</tr>
<tr>
<td valign="top">
<p align="justify">
```

Одной из замечательных особенностей HTML-таблиц по праву считается поддержка многоуровневой вложенности.

```
<br><br>
```

Другими словами, одна таблица может включать другую, та, в свою очередь, ещё одну и т.д.

```
<br><br>
```

Эта особенность, в отличие от других способов представления данных в электронном документе, позволяет более точно размещать отдельные элементы страницы относительно друг друга и границ самого документа, отображаемого браузером.

```
</p>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<!--Левая таблица с текстом. Конец-->
```

```
</td>
```

```
<td valign="top" width="50%">
```

```
<!--Правая таблица со списком. Начало-->
```

```
<table align="center" border="1" cellspacing="3" cellpadding="5" width="100%" >
```

```
<tr>
```

```
<th bgcolor="#000000"><font color="white">Преимущества вложенных таблиц</font></th>
```

```
</tr>
```

```
<tr>
```

```
<td valign="top">
```

```
<ol type="1">
```

```
<li>Гибкая масштабируемость структуры электронного документа в целом<br><br>
```

```
<li>Широкие возможности позиционирования отдельных элементов страницы<br><br>
```

```
<li>Многоуровневое представление разнородных информационных данных<br><br>
```

```
<li>Расширенные оформительские возможности<br><br>
```

```
</ol>
```

```
</td>
```

```
</tr>
```

```

</table>
<!--Правая таблица со списком. Конец-->
</td>
</tr>
</table>
<!--Основная несущая таблица. Конец-->
</body>
</html>

```

Процесс формирования веб-страницы путем компоновки текстовых и графических элементов называется *версткой*. Если в процессе верстки веб-страницы в качестве структурной основы для расположения текстовых и графических элементов используются таблицы, то такая верстка называется *табличной версткой*.

Практикум «Вложенные таблицы. Табличная верстка веб-страниц»

1. Рассмотрите схему предполагаемого размещения материала на веб-странице (рис.?), разбейте ее мысленно на составные части, стараясь выделить основную несущую таблицу и одну или несколько вложенных. Создайте в HTML-документе таблицу согласно приведенной схеме.

Схема размещения информации на веб-странице, реализуемая использованием вложенных таблиц:

| | | | |
|------------------|------------------------------|---------|----------|
| Логотип | Название сайта | | Контакты |
| Меню сайта | Заголовок страницы (раздела) | | |
| | Фото 1 | Текст 1 | |
| | Фото 2 | Текст 2 | |
| Строка копирайта | | | |

2. Найдите в сети Интернет страницу сайта-визитки или промо-сайта, в исходном коде которой «читается» табличная верстка¹⁶. Сохраните архив страницы на свой компьютер. Внося изменения в исходный код сохраненного документа (веб-страницы), определите границы и состав несущей таблицы;

¹⁶ В качестве примеров можно рассмотреть сайты: www.avto-pomosh.ru; www.babaev.ru;

вложенных таблиц. Для облегчения визуального определения границ основных (несущих) и вложенных таблиц, а также состава их строк и ячеек, поочередно вносите в атрибуты тегов <TABLE> сохраненного HTML-документа соответствующие значения ширины и цвета бордюров таблицы. Сформируйте представление о табличном способе верстки веб-страницы.

3. На основе изученного материала и результатов проведенного исследования, создайте заглавную страницу воображаемого промо-сайта способом табличной верстки.

Форма отчета

1. Результат выполнения пункта 1 данной работы демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.
2. Результат выполнения пункта 3 представляется в виде папки, включающей основной HTML-документ и все связанные с ним файлы. Содержимое основного HTML-документа демонстрируется преподавателю: 1) в окне текстового редактора; 2) в окне браузера.

Контрольные вопросы

1. Что представляют собой вложенные таблицы? Назовите преимущества использования вложенных таблиц для веб-дизайна.
2. Что называют версткой сайта? Что представляет собой «табличная верстка» сайта?
3. Для создания сайтов какого типа используется преимущественно табличная верстка и почему?

3.8. Добавление мультимедиа на страницу

Для повышения привлекательности и информативности веб-страницы, на ней, помимо текстовой и изобразительной информации, могут быть размещены звук, видео или анимация, в том числе содержащая интерактивные элементы.

Большинство мультимедийных компонентов, встроенных в веб-страницу (аудио- и видеоролики), представляют собой зависящий от времени материал. Это означает, что процесс проигрывания можно запустить, остановить, «перемотать», повторить и пр. В этой связи, планируя помещение в документ мультимедийного фрагмента, следует иметь представление о способах проигрывания мультимедийных элементов, типах файлов, хранящих мультимедийные данные, их популярности/распространенности среди пользователей, поддержку браузерами и т. п.

Существует огромное количество форматов звуковых файлов, видео и анимации. Большинство из них не поддерживается даже современными браузерами. Ниже представлены основные форматы мультимедиа, которые позволительно встраивать в HTML-документ.

| Формат файла | Тип файла | Расширение |
|--|-------------------------------------|--------------------------------|
| Sun Systems sound | Цифровое аудио | .au |
| Windows sound (формат ОС Microsoft Windows) | Цифровое аудио | .wav |
| Audio Interchange (формат платформы Macintosh) | Цифровое аудио | .aiff |
| MPEG/MP3 audio | Цифровое аудио | .mpg, .mp3 |
| MIDI – Musical Instrument Digital Interface | Команды управления звуком | .mid, .midi fffffffffffffff |
| RealAudio, RealMedia | Аудио/видео поток | .ra, .rm, .ram |
| MPEG video | Видео | .mpg, .mpeg |
| QuickTime | Аудио, видео, а/в поток | .mov |
| Microsoft video | Аудио, видео, а/в поток | .avi |
| Macromedia Shockwave Flash | Аудио, видео, а/в поток Анимация | .swf |

Для встраивания любых мультимедийных файлов в HTML-документ можно использовать два пути:

1. Создание гиперссылки на файл и размещение ее на странице.
2. Применение специального тега вставки мультимедийного файла.

Создание гиперссылки на медиафайл

Правила создания гиперссылки на файлы мультимедийных форматов идентичны оформлению ссылок на HTML-документы или графику. Однако, в данном случае документ, заключенный в файле не может быть напрямую открыт в окне браузера, браузер должен распознать его и загрузить соответствующее вспомогательное приложение. Последнее и выводит мультимедийный файл в надлежащем виде.

Пример ссылки на звуковой файл в формате WAV:

[Поздравление \(1,2 Mb\)](media/music.wav)

При нажатии такой ссылки на компьютере пользователя запускается установленный по умолчанию проигрыватель звуковых файлов – приложение, которое ассоциировано с данным форматом. Если такого приложения нет или оно не настроено, можно просто сохранить файл на жестком диске для будущего использования.

Другим подходом к включению мультимедиа в свои страницы является непосредственное внедрение элементов.

Внедрение мультимедийных элементов

В общем случае внедрение означает резервирование места на странице под включаемый элемент с целью его дальнейшей обработки плагином, который отвечает за его воспроизведение.

Универсальным средством воспроизведения любого медиаконтента (звука, видео, flash-объектов) является запатентованный элемент `<embed>`, который поддерживают большинство браузеров.

Тег `<embed>` имеет ряд параметров, которые позволяют управлять воспроизведением мультимедийного файла непосредственно в окне браузера:

`src` – указание пути к воспроизводимому файлу (обязательный параметр);

`width` – ширина панели воспроизведения (обязательный параметр);

`height` – высота панели воспроизведения (обязательный параметр);

`pluginspage` – указание местонахождения в Сети нужного плагина, если он не установлен на данной системе;

`autostart` – автоматическое начало воспроизведения файла (значение – true).

Значение по умолчанию – false;

`autoload` – запрет автоматической загрузки файла (значение – false);

align – выравнивание панели воспроизведения относительно текста на странице. Возможные значения: left, right, center, top. Значение по умолчанию – baseline;

volume – установка громкости воспроизведения (0-256 или определяется в процентах);

starttime – время начала воспроизведения в минутах и секундах (формат записи "mm:ss"). Значение по умолчанию "00:00";

endtime – время окончания воспроизведения в минутах и секундах (формат записи "mm:ss"). Значение по умолчанию – конец звукового файла;

controls – указание элементов управления на панели воспроизведения. Значения: consol, smallconsole, playbutton, pausebutton, stopbutton, volumelever.

console – отображение полного набора элементов управления на панели воспроизведения;

smallconsole – компактный набор элементов управления на панели воспроизведения;

playbutton – отображение только кнопки воспроизведения;

pausebutton – отображение только кнопки паузы;

loop – воспроизведение повторяется сначала, после его завершения.

Пример конструкции тега **<embed>**:

```
<embed name="Moviel" src="moviel.mov" width="240" height="120" autostart="true" pluginspage="http://www.apple.com/quicktime/download/">
</embed>
```

Альтернативой элементу **<embed>** является тег **<object>**, рекомендованный W3C. Реализовать отображение подключаемого контента с использованием тега **<object>** можно двумя способами:

- 1) используя MIME-тип объекта, чтобы подключить соответствующий плагин для воспроизведения содержимого:

```
<object type="application/x-shockwave-flash" data="myContent.swf" width="400" height="300">
<p>Альтернативный контент</p>
</object >
```


Этот метод не привязан к какому-либо определенному браузеру и поэтому предпочтителен для реализации.

- 2) используя атрибут `classid` у объекта, чтобы браузер смог загрузить необходимый элемент управления ActiveX Flash-плеера.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="400" height="300">
<param name="movie" value="myContent.swf"/>
<p>Альтернативный контент</p>
</object >
```

Этот способ создан специально для Internet Explorer на Windows. Результат его применения зависит от типа браузера.

В приведенных примерах использованы атрибуты:

`type` – MIME-тип файла;

`data` – путь к файлу данных (для объектов, работающих с данными);

`classid` – уникальный регистрационный идентификатор встраиваемого компонента (`clsid` задает уникальный идентификационный номер встраиваемого на страницу элемента управления ActiveX).

Тег `<param>` используется для определения параметров или переменных для элементов `object` или `applet`.

Тег-контейнер `<object>` может включать множество других атрибутов. Дополнительную информацию об использовании данного тега можно найти самостоятельно.

Вставка аудио и видеороликов.

Для добавления аудиофайла на страницу в HTML5 предусмотрен специальный тег `<audio>`.

Базовый код вставки аудио:

```
<audio src="sound.mp3"> </audio>
```

Для обеспечения универсального воспроизведения различными браузерами:

```
<audio>
<source src="sound.mp3">
<source src="sound.aiff">
</audio>
```

Видеофайл в HTML5 добавляется аналогичным образом, с использованием тега `<video>`.

Например: `<video src="film.avi"> </video>`.

Теги `<audio>` и `<video>` могут содержать следующие атрибуты:

`autoplay` – при его добавлении воспроизведение файла начинается сразу же после загрузки страницы;

`controls` – добавляет панель управления;

`loop` – воспроизведение файла повторяется сначала после его завершения;

`preload` – используется для загрузки файла вместе с загрузкой страницы. Игнорируется если использован `autoplay`;

`poster` – (для видео) указывает путь к изображению, пока видео не воспроизводится или недоступно;

также используются атрибуты: `width`, `height`, `type` в значениях, описанных выше.

Ограничения использования тегов `<audio>` и `<video>`.

1. Спецификацией HTML5 не поддерживается воспроизведение потокового мультимедиа¹⁷. В настоящее время отсутствует стандарт, регламентирующий переключение битрейта в элементе `video` в HTML5; текущие реализации поддерживают лишь полные мультимедийные файлы. Однако в спецификации имеются положения, предусматривающие поддержку потокового звука и видео в будущем, как только будет обеспечена поддержка соответствующих форматов.
2. На использование мультимедийной информации распространяются ограничения, связанные с кроссдоменным разделением ресурсов (CORS – cross-origin resource sharing).
3. Полноэкранное видео не может запускаться из сценариев, поскольку предоставление элементу, управляемому сценарием, возможности захвата всего экрана недопустимо из соображений безопасности. Однако браузеры обеспечивают для пользователей возможность выбора

¹⁷ *Потоковое мультимедиа* – мультимедийные данные, непрерывно получаемые от провайдера потокового вещания (например, телевидение)

полноэкранного режима просмотра видео путем предоставления дополнительных элементов управления.

4. Обеспечение доступности элементов audio и video для людей с ограниченными возможностями еще не полностью специфицировано. В настоящее время ведется работа по созданию спецификации WebSRT, регламентирующей поддержку субтитров на основе популярного формата SRT.

3.9. Стилиевое оформление веб-страницы с использованием CSS.

Спецификация языка разметки HTML позволяет разработчику электронных документов изменять внешний вид некоторых элементов страниц. Для этого составляются специальные правила отображения конкретного элемента, называемые каскадными таблицами стилей (CSS Cascading Style Sheets) или стилиевыми шаблонами.

Каскадная – спецификация HTML разрешает использовать для одного и того же элемента несколько стилиевых правил, интерпретируемых браузером последовательно, то есть каскадом;

таблица – формат записи стилиевых правил напоминает табличное представление данных: заголовок таблицы соответствует наименованию элемента, класса или идентификатора стилия; в качестве ячеек и рядов выступают стилиевые свойства и их значения;

стилией – под стилием понимают приведение какого-либо явления к общему набору правил и определений. В этом смысле CSS – способ дополнительного форматирования стандартных тегов HTML.

Пример записи стилия оформления заголовка 3-го уровня:

```
h3 { color: blue; font-family: Tahoma, Verdana, Arial; }
```

или в другом виде:

```
h3  
{  
color: blue;  
font-family: Tahoma, Verdana, Arial;  
}
```

Способы определения таблиц стилей

Любая таблица CSS должна быть интерпретирована браузером для того, чтобы правила CSS, обозначенные для конкретных элементов вступили в силу.

Определение таблиц стилей возможно четырьмя способами:

1. Ссылка на внешний файл

Если стилиевые шаблоны разместить в одном текстовом файле (с расширением .css), то с помощью тега <LINK> из текущего документа можно сделать ссылку на внешний CSS-файл стилиевых шаблонов, например:

```
<link rel="stylesheet" type="text/css" href="style.css">.
```

Конструкция указания пути к файлу находится в пределах раздела HEAD. Браузер, анализируя HTML-код, обратится по указанному пути и, обнаружив указанный файл, отобразит элементы в соответствии с указанным в файле стилем.

В случае ошибки интерпретации HTML-кода браузером, плохой связи с сервером и пр., внешний файл CSS может не загрузиться, вследствие чего стиль для нужных элементов HTML не будет переопределен.

Если внешний файл включает слишком большое количество стилевых шаблонов (что скажется на размере файла), то существует вероятность того, что браузер не сумеет полностью интерпретировать файл CSS или вообще исчерпает лимит времени по загрузке данных. В первом случае стили части элементов не будут переопределены (браузер успеет обнаружить только правила, размещенные в верхней части CSS-файла); во втором случае – все элементы страницы останутся без изменения, загрузившись по умолчанию.

2. Задание стилевой конструкции внутри HTML-страницы

```
< style type="text/css">
<!--!
body { font-family: Tahoma, Verdana, Arial;}
input { background-color: #cecece;}
-->
</style>
```

Данная конструкция также должна присутствовать в разделе HEAD. Описание стиля заключается между символами комментариев с тем, чтобы браузеры, не поддерживающие CSS, пропускали содержание стилевых шаблонов. Браузеры, поддерживающие CSS интерпретируют код и отобразят страницу в соответствии с указаниями.

Наличие слишком большого количества шаблонов CSS заметно увеличивает размер страницы, что скажется на времени загрузки документа в браузере.

3. Включение в теговую конструкцию отдельного HTML-элемента

```
<p align="justify" style="color: #00ffcc; font-family: Tahoma, Verdana, Arial;">
...текст параграфа...
</p>
```

4. Импортрование стилевого шаблона (ссылка на внешний файл в Интернет)

```
< style type="text/css">
```

```
<!--!  
@import: url (style.css);  
-->  
</style>
```

При импортировании стилевого шаблона, оператор импорта стиля должен предшествовать всем прочим описателям стилей:

```
<style>  
@import: url(http://kuku.ru/style.css)  
a {color:cyan;text-decoration:underline;}  
</style>
```

Все четыре способа определения стилевого шаблона можно использовать одновременно в пределах одного HTML-документа.

При сочетании различных способов следует учитывать особенности браузеров.

Правила составления стилевых шаблонов

Любое правило оформления стиля состоит из *селектора* и *определения шаблона*.

Селектор – это условие отображения стиля для заданного элемента или наименование внутреннего класса (идентификатора) для него.

Определение шаблона – это описание стилевых правил оформления, которые чередуются через точку с запятой и заключаются в фигурные скобки.

| | |
|----------|---|
| Селектор | Определение (определят правило отображения цвета заголовка и правило отображения гарнитуры) |
|----------|---|

```
H3 { color: blue; font-family: Tahoma, Verdana, Arial; }
```

Возможно наследование стилевых правил для нескольких селекторов одновременно, например:

```
td, th, p, h3, div {color: blue; font-family: Tahoma, Verdana, Arial; font-size: 10px;}
```

В качестве селектора CSS могут выступать:

- *элементы HTML* (будет определяться стиль для конкретного элемента страницы):

```
body { color: blue; }
```

Весь текст в пределах BODY будет голубым, но при добавлении, например, таблицы, назначение стилевого шаблона пропадает для текста внутри ячеек.

- *Классы* (использование классов позволяет переопределять стиль для любого элемента, которому присвоен данный класс).

Наименование класса начинается с точки и пишется строчными буквами, например:

```
.reg { color: blue; }
```

В этом случае все элементы, которым присвоен класс `.reg` будут менять цвет на голубой:

```
<font class="reg"> ... </font>
```

Если дополнить селектор класса наименованием конкретного элемента, то действие стилевого правила будет распространяться только на этот элемент:

```
hr.reg { color: blue; }
```

- *Идентификаторы* (использование классов позволяет переопределять стиль для любого элемента так же, как использование классов. Однако селектор идентификатора часто применяется для задания уникального имени элементу, который задействован в программном сценарии (скрипте)).

Запись идентификатора начинается с символа # и заканчивается наименованием.

```
#reg { color: blue; }
```

В этом случае элемент, которому присвоен идентификатор `reg` будут менять цвет на голубой.

```
<td id="reg"> ... </td >
```

Псевдоклассы

Псевдоклассами называют определенные условия форматирования в соответствии с которыми браузер подставляет необходимые стилевые правила отображения данных. При этом в исходной структуре документа такие классы не присутствуют: они создаются в процессе интерпретации HTML-кода браузером.

Псевдоклассы могут использоваться для задания различных типов форматирования по нескольким разновидностям элементов.

Например, если рассматривать гиперссылку, то её отображение, в зависимости от состояния (активная, посещённая, непосещённая, наведён курсор) может быть различным. Цвета отображения состояния ссылки можно прописать в теге BODY, а можно задать псевдоклассами при записи в стилевом шаблоне:

`a: link { color: blue; }` – отображение непосещенной ссылки;

`a: active { color: red; text-decoration: underline; }` – отображение активной ссылки;

`a: visited { color: gray; }` – отображение посещенной ссылки;

`a: hover { color: green; }` – вид ссылки при наведении курсора мыши.

Если есть необходимость визуально выделить одни ссылки относительно других, наряду с псевдоклассами используют обычные селекторы классов. Например:

`a: link.red { color: blue; }`;

`a: active.red { color: red; text-decoration: underline; }`;

`a: visited.white { color: gray; }`;

`a: hover.white { color: green; }`.

Наследование и переопределение стилей.

1. Сначала применяются стили умолчания браузера
2. Стили умолчания браузера переопределяются прилинкованными стилями (элемент LINK заголовка документа).
3. Прилинкованные стили переопределяются описаниями стилей в элементе STYLE
4. Стили элемента STYLE переопределяются атрибутом STYLE в любом из элементов разметки

3.10. Текстовые блоки. Верстка веб-страницы блоками <div>

При верстке сайтов с многоколончатый макетом, применяют, так называемую, блочную верстку.

Основа блочной системы – это тег `<div>...</div>`, который является контейнером для контента. Внутри него также могут содержаться другие контейнеры `<div>...</div>`.

Информация, размещенная в блоке, прежде всего, должна занять свое, определенное компоновкой, место на веб-странице. Для этого необходимо указать *положение блока*. Кроме того, информация должна быть отображена соответствующим дизайну сайта образом. Для этого требуется указать *стилевое оформление* блока и его содержимого.

Свойства положения и оформления блока удобно задаются через описание таблицы стилей. Для этого в теге `<div>` указываются атрибуты: 1) `style` или 2) `class` или `id` с именем селектора.

Например:

1. `<div style="color: gray;">` Фрагмент текста `</div>`.
- 2.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Тег DIV</title>
    <style type="text/css">
      .block1 {
        width: 200px;
        background: #ccc;
        padding: 5px;
        padding-right: 20px;
        border: solid 1px black;
        float: left;
      }
      #block2 {
        width: 200px;
        background: #fc0;
        padding: 5px;
        border: solid 1px black;
        float: left;
        position: relative;
        top: 40px;
        left: -70px;
      }
    </style>
  </head>
  <body>
    <div class="block1">Текст 1.</div>
    <div id="block2">Текст 2.</div>
  </body>
</html>
```

Задание положения блока.

Атрибут **float** определяет плавающий блок текста. Он может принимать значения:

left – блок прижат к левой границе охватывающего блок элемента;

right – блок прижат к правой границе охватывающего блок элемента;

none – элемент страницы не перемещается, будет отображен там, где он должен быть (значение по умолчанию);

both – текст может обтекать блок с обеих сторон.

Совместно с **float** очень часто используется стилевое свойство **clear**, которое управляет потоком после перемещаемых блоков. Стилевое свойство **clear** задает, какие границы элемента страницы не могут совпадать с предшествующим перемещаемым блоком свойством **float**.

Свойство **clear** может принимать значения

left – элемент страницы устанавливается ниже любого предыдущего, перемещенного влево блока;

right – элемент страницы устанавливается ниже любого предыдущего, перемещенного вправо блока;

both – элемент страницы устанавливается ниже любого предыдущего перемещенного блока;

none – ограничений на положение блока относительно перемещаемых блоков не накладывается.

Верстка блоками DIV не обходится без управления положением HTML-элементов с помощью стилевого свойства **position**, которое определяет схему позиционирования блока. Свойство **position** может принимать значения:

static – задает обычный блок, который не позиционируется индивидуально, а позиционируется в обычном потоке. Свойства **top**, **left**, **bottom** и **right** не учитываются. Является значением по умолчанию;

relative – задает относительное позиционирование блока. Заключается в следующем. Сначала положение блока рассчитывается как при обычном потоке (то есть не берется во внимание относительное позиционирование). Затем блок смещается относительно своего нормального положения в

соответствии со значениями свойств `top`, `left`, `bottom` и `right`. Последующее содержимое выводится в поток так, как будто блок, относительно позиционированный, не был смещен;

absolute – задает абсолютное позиционирование блока. Абсолютно позиционируемые блоки изымаются из общего потока и не влияют на выводимое последующее содержимое. Положение абсолютно позиционируемого блока задается свойствами `top`, `left`, `bottom` и `right` относительно родительского элемента (блока);

fixed – задает фиксированное позиционирование блока. При фиксированном позиционировании положение блока рассчитывается, как и при абсолютном позиционировании, но в дальнейшем блок фиксируется и относительно окна браузера и не прокручивается вместе с остальным содержимым страницы. Данное свойство поддерживается браузерами Opera и Firefox.

Для задания положения блоков, у которых свойство `position` имеет значение `relative`, `absolute` или `fixed`, используются свойства `top`, `left`, `bottom` и `right`. Познакомимся с ними.

top – задает смещение верхней границы позиционируемого блока вниз относительно верхней границы родительского элемента;

left – задает смещение левой границы позиционируемого блока вправо относительно левой границы родительского элемента;

bottom – задает смещение нижней границы позиционируемого блока вверх относительно нижней границы родительского элемента;

right – задает смещение правой границы позиционируемого блока влево относительно правой границы родительского элемента.

Задавать величины перечисленных выше свойств можно в абсолютных значениях (например, пиксели), в процентных величинах относительно размеров родительского элемента, и также могут принимать значение `auto` (является значением по умолчанию).

При абсолютном, относительном и фиксированном позиционировании блоки могут накладываться друг на друга, и в этом случае может появиться вопрос, какой из них окажется «выше»? То есть, какой элемент будет отображаться, а какой будет перекрываться находящимся «выше» блоком. Решить этот вопрос поможет свойство `z-index`. Оно определяет позицию уровня, которую будет занимать блок. Его значением может быть любое положительное или

отрицательное целое число. При этом элементы с большим значением **z-index** будут перекрывать блоки с меньшим значением этого свойства.

Для иллюстрации описанных выше свойств рассмотрим пример.

```
<div style="background:#A485B8;">1<br>text</div>
<div id="dopoln_block">
<div style="background:#66AAD7; width:200px;">2<br>текст</div>
</div>
<div style="background:#C6C600;">3<br>содержимое</div>
```

Для ознакомления с относительным позиционированием добавим в один блок свойства относительного позиционирования

```
<div style="background:#A485B8;">1<br>text</div>
<div id="dopoln_block">
<div style="background:#66AAD7; width:200px; position: relative; top:10px;
left:10px;">2<br>текст</div>
</div>
<div style="background:#C6C600;">3<br>содержимое</div>
```

Для ознакомления с абсолютным позиционированием посмотрите, как отобразится следующий код в браузере

```
<div style="background:#A485B8;">1<br>text</div>
<div style="position:relative;" id="dopoln_block">
<div style="background:#66AAD7; width:200px; position: absolute; top:10px;
left:10px;">2<br>текст</div>
</div>
<div style="background:#C6C600;">3<br>содержимое</div>
```

В последнем примере, для того чтобы положение абсолютно позиционируемого блока можно было задавать свойствами **top** и **left**.

Задание свойств стилевого оформления блока и его содержимого приведены в приложении 4.

3.11. Проверка валидности HTML и CSS кода веб-страниц

Исходный код страницы должен быть написан по правилам веб-стандартов и правилам языка разметки. Ошибки, допущенные в коде, могут повлиять на отображение ресурса в браузере и плохо сказаться на оценке ресурса со стороны информационно-поисковых систем. Проверка веб-кода на валидность – это проверка его на соответствие стандартам и сертификатам W3C (от англ. World Wide Web Consortium, W3C).

W3C – это технические законодатели Сети, которые разрабатывают стандарты и правила для написания кода. Сертификаты и стандарты W3C обязательны к исполнению для всех, кто работает в Сети. Единые стандарты в правописании кода нужны для того, чтобы все Сетевые приложения общались в едином языковом пространстве, на стандартных языках, и понимали друг друга во время работы с веб-документами. W3C не только создает Сетевые стандарты, но и активно способствует в их соблюдении. W3C имеет онлайн-сервисы для проверки кода HTML/XHTML и CSS на валидность.

Для проверки веб-страниц на наличие ошибок и замечаний существует множество путей и способов. Условно они делятся на онлайн-овые и локальные. Онлайн-овые предназначены для проверки страниц с помощью браузера через Интернет, а локальные используются для проверки документов на текущем компьютере.

Код опубликованного в Интернете сайта можно проверить на валидность, используя бесплатные онлайн-сервисы от W3C. Прежде всего, это сервис, доступный по адресу: <http://validator.w3.org/> – для проверки html-кода.

Валидаторы от W3C имеют интуитивно понятный интерфейс. Работать с ними просто. Сервисы дают возможность проводить проверку в трех режимах и имеют, соответственно, всего три кнопки:

- проверить URL (для проверки нужно указать адрес любой страницы сайта, доступного в Сети);
- проверить загруженный файл (для проверки нужно указать путь к проверяемому файлу);
- проверить набранный текст (для проверки нужно скопировать и вставить в окно валидатора проверяемый код).

Последние два способа особенно полезны при проверке веб-документов или текстов, находящихся на локальных компьютерах.

Сервисы от W3C проверяют html-код на валидность, и сразу указывают на ошибки, если таковые имеются. Каждая ошибка комментируется (комментарии на английском языке).

Валидатор производит несколько проверок кода. Основные из них:

1. *Валидация синтаксиса* – проверка на наличие синтаксических ошибок. `<foo bar="baz">` является корректным синтаксисом, несмотря на то, что `<foo>` не является допустимым HTML-тэгом, так что проверка синтаксиса является минимально полезной для написания хорошего HTML.
2. *Проверка вложенности тэгов* – тэги должны быть закрыты в обратном порядке относительно их открытия. Например, эта проверка обнаруживает ошибки с неправильно закрытыми `<div>`.
3. *Валидация DTD* – проверка соответствия Вашего кода указанному Document Type Definition. Она включает проверку названий тэгов, атрибутов, и «встраивания» тэгов (тэги одного типа внутри тэгов другого типа).
4. *Проверка на посторонние элементы* – проверка выявляет все, что есть в коде, но отсутствует в DTD. Например, пользовательские тэги и атрибуты.

Сервис от W3C для проверки CSS-кода доступен по адресу: <http://jigsaw.w3.org/css-validator/>.

Все ошибки валидации можно разделить на группы:

- *ошибки в файлах шаблона*. Их не сложно найти и исправить. Если, какие то из мелких ошибок помогают сделать сайт более функциональным или быстрым, их можно оставить;
- *ошибки в сторонних скриптах, подключенных на сайте*. Например, виджет Вконтакте, скрипт Твиттера или видео-файлы с ютуб. Исправить их никак не удастся, поскольку эти файлы и скрипты находятся на других сайтах и у нас нет к ним доступа;
- *CSS-правила, которые валидатор не понимает* и поэтому считает их ошибками. Валидатор проверяет соответствие кода сайта определенной версии HTML или CSS. Если в шаблоне использованы правила CSS 3, а валидатор проверяет на соответствие версии 2.1, то все правила CSS 3 он будет считать ошибками, хотя на самом деле они таковыми не являются;

- *ошибки, которые осознанно допускает разработчик*, чтобы получить требуемый результат. Например, теги noindex. Они не валидны, но очень нужны и с этим приходится мириться.
- *«хаки»*. Чтобы получить корректное отображение сайта в некоторых браузерах, иногда, приходится использовать «хаки» – код, который понимает только этот браузер.
- *ошибки самого валидатора*. Бывает, что валидатор не видит каких-то тегов (например, закрывающих) и сообщает об ошибке там, где ее нет.

Таким образом, на работающем сайте практически всегда будут найдены какие-то ошибки валидации. Причем, этих ошибок может быть много. Например, главные страницы Google, Яндекса и mail.ru содержат по несколько десятков ошибок, но эти ошибки не ломают отображение сайтов в браузерах и не мешают этим сайтам работать.

Практикум «Проверка валидности HTML и CSS кода веб-страниц»

1. Найдите онлайн-ресурсы, выполняющие проверку валидности html- и css-кода. Выберите среди множества ресурсов три, которыми будете пользоваться.
2. Выполните проверку валидности html- и css-кода:
 - а) интернет-представительства крупной компании (3 организаций);
 - б) сайта-визитки или промо-сайта (3 образца);
 - б) персональной страницы (3 образца).

Для каждого ресурса выполните проверку каждым из валидаторов. Сравните результаты.

3. Сделайте вывод о качестве кода протестированных ресурсов; о работе валидаторов.

Форма отчета

Результат выполнения работы оформите в виде текстового документа, содержащего:

- перечень валидаторов, выбранных для проведения тестирования;
- перечень сайтов, выбранных для анализа, с указанием полного названия, характеристики и url-адреса ресурсов;
- результаты проверки валидности кода;
- вывод, сделанный по результатам тестирования.

Контрольные вопросы

1. Зачем требуется выполнять проверку валидности кода веб-страницы?
2. Назовите 1-2 ресурса, осуществляющих проверку валидности кода веб-страниц.
3. Какие ошибки ищет валидатор при тестировании html-кода?
4. Насколько критично наличие ошибок в html-коде, выявленных валидатором. Какие сайты содержат минимальное, а какие наибольшее число ошибок?

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Вроблевски Люк. Сначала мобильные [Текст]/Л. Вроблевски -М: Изд-во «Манн, Иванов и Фербер», 2012. –176с.
2. Голомбински К., Хаген Р. Добавь воздуха! Основы визуального дизайна для графики, веб и мультимедиа. – СПб.: Питер, 2013. – 272 с.
3. Квинт И. HTML, XHTML и CSS на 100%. – СПб.: Питер, 2010. – 384 с.
4. Макнейл П. Веб-дизайн: идеи, секреты, советы. Самые актуальные темы, тенденции, стили. Питер, 2012. – 271 с.
5. Маркотт И. Отзывчивый веб-дизайн [Текст]/И. Маркотт -М: Изд-во «Манн, Иванов и Фербер», 2012. -170с.
6. Нильсен Я., Перниче К. Веб-дизайн: анализ удобства использования веб-сайтов по движению глаз.: перев. с англ. – М.: ООО «И. Д. Вильямс», 2010. – 480 с.
7. Петюшкин А. В. HTML в веб-дизайне. – СПб.: БХВ-Петербург, 2004. – 400 с.
8. Пилгрим М. Погружение в HTML5: перев. с англ. – СПб.: БХВ-Петербург, 2011. – 304 с.
9. Седерхольм Д. Пуленепробиваемый веб-дизайн. Библиотека специалиста. 3-е изд. – СПб.: Питер, 2012. – 304 с.
10. Серго А. Г. Доменные имена: монография. – М.: Российская государственная академия интеллектуальной собственности (ФГБОУ ВПО РГАИС), 2013. – 312 с.

ПРИЛОЖЕНИЯ

Приложение 1

Декларация типа документа

Рекомендуется начинать любой HTML-документ со строки, содержащей декларацию типа документа (DTD, document type declaration).

HTML 4.0 поддерживает три типа таких деклараций:

документ *строго соответствует* стандарту HTML 4.0, т. е. не содержит ни морально устаревших элементов, ни фреймов. Декларация имеет вид:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

документ является *переходным* к стандарту HTML 4.0, т. е. может содержать морально устаревшие элементы. Декларация имеет вид:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

документ *содержит фреймы*. Декларация имеет вид:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
```

```
"http://www.w3.org/TR/html4/frameset.dtd">
```

Строка декларации документа, созданного в соответствии с HTML 5.0

```
<!DOCTYPE HTML>
```

Дополнительно: http://programmerbook.ru/html/doctype/ru_html5/

Вид веб-страницы словаря терминов предметной области «Веб-дизайн»

Глоссарий. Веб-дизайн

[Web](#)
[Веб-дизайн](#)
[Веб-страница](#)
[Веб-сайт](#)
[Дизайн](#)

Определения WEB

- **Web** - система доступа к связанным между собой документам на различных компьютерах, подключённых к Интернету. [Источник](#)
- **Web** (с англ. - паутина, сеть, система. *Синонимы: веб, Всемирная паутина, «WWW»*) - глобальное информационное пространство, основанное на физической инфраструктуре Интернета и протоколе передачи данных http. [Источник](#)
- **Web** - «всемирная паутина», основная служба в сети интернет, организованная на принципах гиперсреды, позволяющая получать доступ к информации на любых серверах, подключенных к сети. [Источник](#)

[Вверх](#)

Определения Веб-дизайна

- **Веб-дизайн** (англ. *web - паутина, сеть и design - оформление, проектирование*), дизайн сайтов и их отдельных элементов, создание графических рекламных материалов в Интернете, а также проектирование его структуры, навигации и программ, нужных для работы сайта, то есть создание сайта целиком. Обычно веб-дизайн осуществляют специализированные студии. [Источник](#)
- **Веб-дизайн** - частный случай промышленного дизайна. Вид художественно-проектной деятельности, направленный на создание и обеспечение удобства использования веб-ресурсов. [Источник](#)
- **Веб-дизайн** - это весь процесс создания сайта. Включает в себя «внешний вид», систему навигации (переходы по ссылкам), программы для функционирования сайта. Это проектирование практичной, информативной и удобной для пользователя системы. [Источник](#)

[Вверх](#)

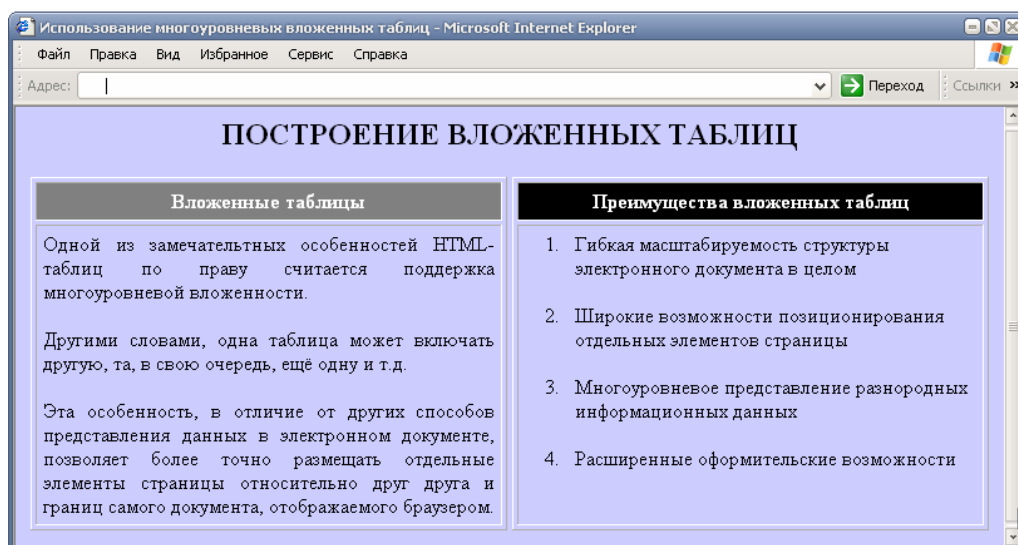
Определения Веб-страницы

- **Веб-страница** - самостоятельная часть веб-сайта, отдельный документ в Интернете, снабженный уникальным адресом (URL), который может содержать текст, графические, звуковые, видео файлы, анимацию. Пишется на языке HTML. [Источник](#)
- **Web-страницы или гипертекстовые документы** (html-документы) - это текстовые файлы, размеченные тегами с помощью языка программирования, он является языком разметки или средством форматирования гипертекста.

www.onvolga.ru/infoteka/terms.html

Табличная верстка веб-страниц

Пример организации информации с использованием вложенных таблиц



Вид страницы промо-сайта, сверстанного на основе таблиц



Области применения стилевых шаблонов

Единицы измерения в таблицах стилей

| Абсолютные | Относительные |
|----------------------|-----------------------------|
| In - дюйм≈2,5 см | em - высота шрифта элемента |
| mm - миллиметр | ex - высота буквы x |
| cm - сантиметр | px - пиксел |
| pt - пункт≈1/7 дюйма | % - процентное соотношение |
| pc - пика=12 пунктов | |

Свойства форматирования текста

| Свойство | Пример записи | Функция |
|------------------------|---|--|
| font-family | font-family:Tahoma, Arial; | выбор гарнитуры для отображения |
| font-size | font-size: 24px; font-size: 12pt; font-size: 120%; font-size: large; | размер шрифта |
| font-style | font-style: normal; font-style: italic; | выбор прямого или наклонного начертания |
| font-weight | font-weight: bold; | наличие жирного начертания |
| font-variant | font-variant: small-caps; | преобразование строчных букв в прописные |
| text-decoration | text-decoration:underline; | подчеркивание текста |
| text-align | text-align: right; | тип выравнивания текста |
| text-transform | text-transform: uppercase; | выбор регистра букв |
| letter-spacing | letter-spacing: 1em; letter-spacing: 10pt; | межбуквенный интервал |
| line-height | line-height: 5mm; | межстрочный интервал |
| color | color: white; color: #003366; | цвет текста |
| background | background-color:#003366; | цвет заднего фона текста |

Свойства структурного форматирования

| Свойство | Пример записи | Функция |
|-------------------------|--|--|
| border-width | border-width: 20px; | ширина границы структурного элемента |
| border-style | border-style : solid; border-style : dotted; | тип декоративного отображения границы элемента |
| border-color | border-color: gray; | цвет границы структурного элемента |
| width | width: 300px | ширина структурного элемента |
| height | height: 120px | высота структурного элемента |
| background-color | background-color:#CECECE; | цвет заднего фона структурного элемента |
| margin | margin: 15px 5px 50px 50px; margin-left:50px; margin-right:5px; margin-top:15px; margin-bottom:50px; | Определение размера поля относительно верхнего, правого, нижнего, левого структурного элемента |
| padding | padding-top: 10em; padding-right: 25px; | отступ от краев структурного элемента |
| float | float: left; | расположение структурного элемента относительно других |