

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
КАФЕДРА КОМПЬЮТЕРНОЙ ИНЖЕНЕРИИ



МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к самостоятельной работе по дисциплине

«ИНТЕРНЕТ-ТЕХНОЛОГИИ»

для студентов уровня профессионального образования «магистр»

всех направлений подготовки и форм обучения

Донецк
2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
КАФЕДРА КОМПЬЮТЕРНОЙ ИНЖЕНЕРИИ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к самостоятельной работе по дисциплине

«ИНТЕРНЕТ-ТЕХНОЛОГИИ»

для студентов уровня профессионального образования «магистр»

всех направлений подготовки и форм обучения

РАССМОТРЕНО

на заседании кафедры

«Компьютерная инженерия»

Протокол № 6 от 20.01.2020 г.

УТВЕРЖДЕНО

на заседании Учебно-издательского

совета ГОУВПО «ДОННТУ»

Протокол № 3 от 27.04.2020 г.

Донецк
2020

УДК 004.738.5(076)
ББК 32.973.202я73
М54

Рецензент:

Зори Сергей Анатольевич – доктор технических наук, доцент, заведующий кафедрой «Программная инженерия».

Составители:

Аноприенко Александр Яковлевич – кандидат технических наук, заведующий кафедрой «Компьютерная инженерия»;

Иваница Сергей Васильевич – кандидат технических наук, доцент кафедры «Компьютерная инженерия»;

Сидоров Константин Андреевич – ассистент кафедры «Компьютерная инженерия».

М54 **Методические указания** к самостоятельной работе по дисциплине «**Интернет-технологии**» [Электронный ресурс] : для студентов уровня профессионального образования «магистр» всех направлений подготовки и форм обучения / ГОУВПО «ДОННТУ», Каф. компьютерной инженерии ; сост.: А. Я. Аноприенко, С. В. Иваница, К. А. Сидоров. — Донецк: ДОННТУ, 2020. — 58 с. — Систем. требования: Acrobat Reader. — Загл. с титул. экрана.

Целью освоения теоретической части курса «Интернет-технологии» является как изучение и эффективное использование современного программного обеспечения для просмотра и верстки сайтов, так и рассмотрение вопросов, связанных с перспективами развития информационных возможностей Интернет.

В методические указания к выполнению самостоятельной работы включен материал о развитии гипертекстовых технологий и появившегося на их основе языка гипертекстовой разметки, а также материал о каскадных таблицах стилей.

УДК 004.738.5(076)
ББК 32.973.202я73

Содержание

	стр.
1. Введение: цель и задачи курса	4
2. Развитие технологии гипертекста	6
3. Историческое развитие гипертекста	10
4. Браузеры. История развития	19
5. Развитие языка гипертекстовой разметки HTML	28
6. Язык гипертекстовой разметки	34
7. Технология «Клиент-Сервер»	38
8. Обработка веб-документов в браузере	40
9. Структура документа HTML	45
10. Каскадные таблицы стилей. История развития CSS	49
11. Контрольные вопросы	53

1. Введение: цель и задачи курса

Целью курса «Интернет-технологии» является приобретение теоретических и практических знаний, умений и навыков, ориентированных на эффективное профессиональное использование современных Интернет-технологий.

При этом решаются следующие задачи:

1. Использование информационных ресурсов Интернет для углубленного многоязычного поиска научной и технической информации по теме выпускной работы.
2. Разработка (в основном на основе проведенного поиска и систематизации информации по теме выпускной работы) и размещение на портале магистров ДонНТУ комплексного персонального сайта, ориентированного на тематику выпускной работы и содержащего различные тематические разделы.
3. Освоение в процессе работы над сайтом основ эффективного использования языка гипертекстовой разметки HTML и различных средств и способов обработки и представления графической информации.
4. Приобретение навыков в области проведения и документирования исследовательской работы на базе Интернет-технологий, а также – в области составления тематических электронных библиотек и перечней ссылок.
5. Приобретение знаний и навыков по продвижению в Интернет собственных ресурсов.

Курс «Интернет-технологии» включает в себя 3 основных вида учебных занятий: лекции, лабораторные занятия и самостоятельную работу, основным содержанием которой является разработка информационного наполнения персонального тематического сайта.

Выполнение лабораторных работ предполагает активное использование Интернет для поиска и сбора информации в процессе поэтапной разработки тематического сайта и его размещения на портале магистров. *Систематическая работа над сайтом яв-*

ляется одним из главных условий его качественной реализации и, соответственно, успешного завершения данного курса. В качестве основного источника дополнительной информации по различным вопросам данного курса также рассматриваются в первую очередь информационные ресурсы Интернет. В то же время, для выполнения основного объема самостоятельной работы по данному курсу доступ к интернету не является обязательным условием.

Для успешной самостоятельной работы необходимы:

- систематическое изучение лекционного материала и рекомендованной литературы;
- эффективное использование учебного времени, в том числе доступа к интернету, во время проведения лабораторных работ;
- наличие компьютерного рабочего места с набором рекомендованных браузеров (Chrome, Opera, Firefox, Internet Explorer), графических редакторов и другого, как правило, относительно простого и свободно распространяемого программного обеспечения (соответствующий комплект программного обеспечения содержится в соответствующем разделе портала магистров);
- наличие шаблона, определяющего базовые требования к основным разделам и элементам сайта магистра.

2. Развитие технологии гипертекста

Важнейшие свойства и особенно реализации гипертекста прочно вошли в повседневную жизнь в виду глобализации веб-технологий. Однако гипертекст не всегда был выражен теми технологическими реализациями, которые сегодня кажутся естественными и вполне очевидными. В данном разделе рассматривается эволюция гипертекста в историко-технологическом контексте.

Гипертекст (англ. *hypertext*) — специальным образом организованный текст, позволяющий пользователю осуществлять переход к связанным ресурсам по указателям (ссылкам).

Гипертекстовая система — информационная система, способная хранить информацию в виде электронного текста, позволяющая устанавливать электронные связи между любыми «информационными единицами», хранящимися в ее памяти.

Иными словами, гипертекст представляет такую организацию текстовой информации, внутри которой установлены смысловые связи между ее различными фрагментами. Такие связи называют *гиперсвязями*, а место в тексте, обеспечивающее гиперсвязь, — **гипертекстовой ссылкой (гиперссылкой)**.

В настоящее время в компьютерной сфере термин «*гипертекст*» относят в равной мере к разным объектам:

1) так называют особый метод построения информационных систем, обеспечивающих прямой доступ к данным с сохранением естественных (или специальным образом предустановленных) логических связей между ними;

2) это определенная система представления гипермедийной информации (текстовой и мультимедийной) в виде сети связанных между собой информационных единиц;

3) это особый универсальный компьютерный программный интерфейс, отличительными чертами которого являются его интерактивность и чрезвычайно высокая степень адаптивности по отношению к психологии пользователя.

Гиперссылка в рамках Интернет-технологий представляет собой фрагмент документа HTML и его базовый элемент:

– указывающий на другой файл, который может быть расположен в Интернете;

– содержащая полный путь (URL-адрес) к этому файлу.

Основная идея гипертекстовых систем заключается в концепции автоматически поддерживаемых связей между различными фрагментами информации (информационными единицами). Поддержка таких связей позволяет организовывать «нелинейные» информационные структуры. **В качестве гиперссылок в электронном тексте** могут выступать:

- ссылки на словарь терминов и понятий (выделение ключевых слов в тексте);
- ссылки на персоналии (портреты и краткие биографические сведения);
- ссылки на статические иллюстрации (изобразительные и условно-графические, в т. ч. схемы, таблицы и т. д.);
- ссылки на мультимедийные элементы (анимации, аудио- и видеофрагменты);
- ссылки на хрестоматийный или дополнительный материалы;
- ссылки на структурные элементы текста (оглавление, номер темы, пункт и подпункт, список вопросов для закрепления и устных развернутых ответов и др.);
- ссылки на список монографий, учебной и научной литературы (приводится в конце темы или всего курса);
- ссылки на список организаций;
- ссылки на список исторических событий или дат (хронологический указатель);
- ссылки на список географических названий;
- ссылки на Интернет-ресурсы (образовательные сайты, электронные библиотеки, мультимедийные приложения и др.).

При создании гипертекста, гиперссылки, как правило, выделяются из общей массы текста. Для этого можно использовать следующие приемы:

- изменение цвета гиперссылки относительно общего цвета текста;
- изменение начертания шрифта для гиперссылки;
- увеличение или уменьшение размера шрифта;
- выделение подчеркиванием, курсивом или жирным шрифтом;
- применение различных комбинаций вышеперечисленных способов.

В разнообразных *текстовых редакторах* и программах при создании веб-страниц гиперссылки выделяются различными способами. Например, в программных продуктах MS Office ссылки автоматически выделяются синим цветом и подчеркиваются.

Информационные гипертекстовые узлы обычно представляют собой некое концептуальное утверждение или выражают одну определенную идею. Эта информация может быть по-разному оформлена и структурирована в зависимости от своей функциональной семантики (например, утверждение, разъяснение, справочная информация, развивающая и детализирующая то или иное утверждение, иллюстративная информация, экспериментальные результаты, наблюдения, фактическая информация и выводы, информация прикладного характера и ее семантико-синтаксическое описание).

Формализация структуры гипертекста обычно осуществляется при помощи отображения ее в виде ориентированного графа, в котором точками обозначаются гипертекстовые (информационные) узлы (англ. *hypernode*), а стрелками — связи между ними (гиперссылки). Пример такого изображения в сравнении с более традиционной («древовидной») схемой отображения информации представлен на рис. 1.

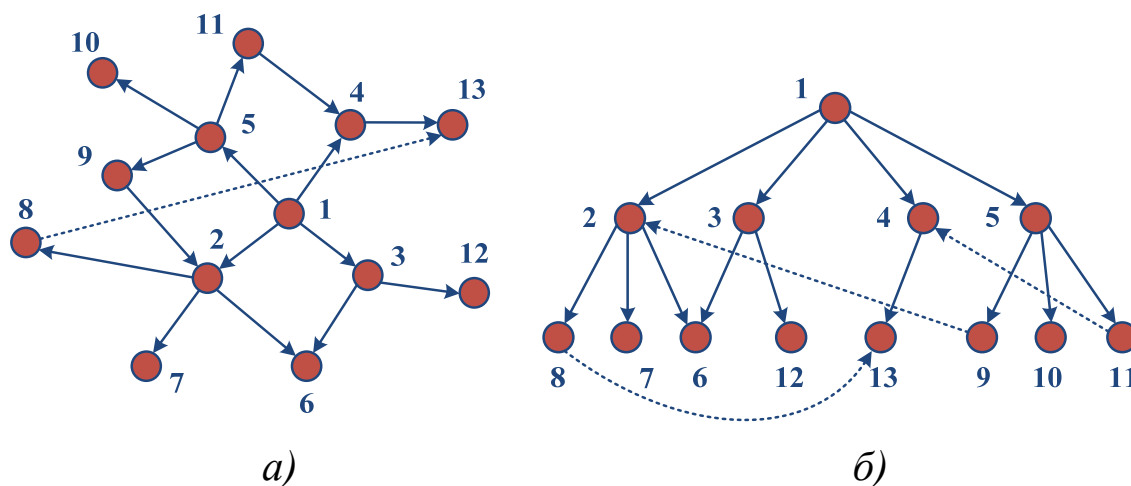


Рис. 1. Схематичное представление гипертекстового пространства в виде ориентированного графа (а) и дерева зависимостей (б)

Информационные узлы образуют **единое гипертекстовое пространство** благодаря связям между ними, которые устанавливаются создателями гипертекста априорно в соответствии с не-

которыми собственными концептуальными представлениями через систему гиперссылок. Тот узел, который является исходным для некоторой ссылки, называется **узлом-адресантом**, или **референциальным информационным узлом**. Узел, в который ведет та или иная ссылка, называется **узлом-адресатом ссылки**, или **референтом (сигнификативным) узлом**. Любой фрагмент информационного узла может быть связан с другим узлом, его частью или со своим любым другим фрагментом с помощью так называемых **якорей** (англ. *anchor*), или специальных не видимых читателю **маркеров**, которые помещаются в определенных частях текста (рисунка, схемы, аудио-, видеофайла и т. п.).

Содержимое информационного узла или его фрагмента выводится на экран компьютера с *помощью активизации гиперссылки*. Они могут быть:

- **однонаправленные** — ведущие из одного гипертекстового узла в другой;
- **двунаправленные** — обеспечивающие перемещение между двумя связанными информационными фрагментами в обе стороны.

До повсеместного распространения браузеров это различие между типами гиперссылок очень активно использовалось в обучающих компьютерных системах. В настоящий момент *все гиперссылки в сети Интернет являются двунаправленными*, так как возможность возврата на любой из нескольких предыдущих этапов просмотра веб-страницы заложена в самом браузере.

Гиперссылки могут быть также классифицированы *в зависимости от их структурно-функциональной роли в общей системе гипертекстового пространства*:

- **референциальные ссылки** для установления отношений перекрестного цитирования;
- **иерархические ссылки** для установления отношений типа «родитель — ребенок».

Эта классификация в настоящее время активно используется для наиболее эффективной системы организации информационного поиска в Интернете во всех наиболее известных поисковых системах.

3. Историческое развитие гипертекста

Следует отметить, что теоретические предпосылки для создания гипертекста были заложены еще за полвека до создания первого сайта. В хронологическом контексте историю гипертекста можно условно разделить на три поколения (рис. 2).

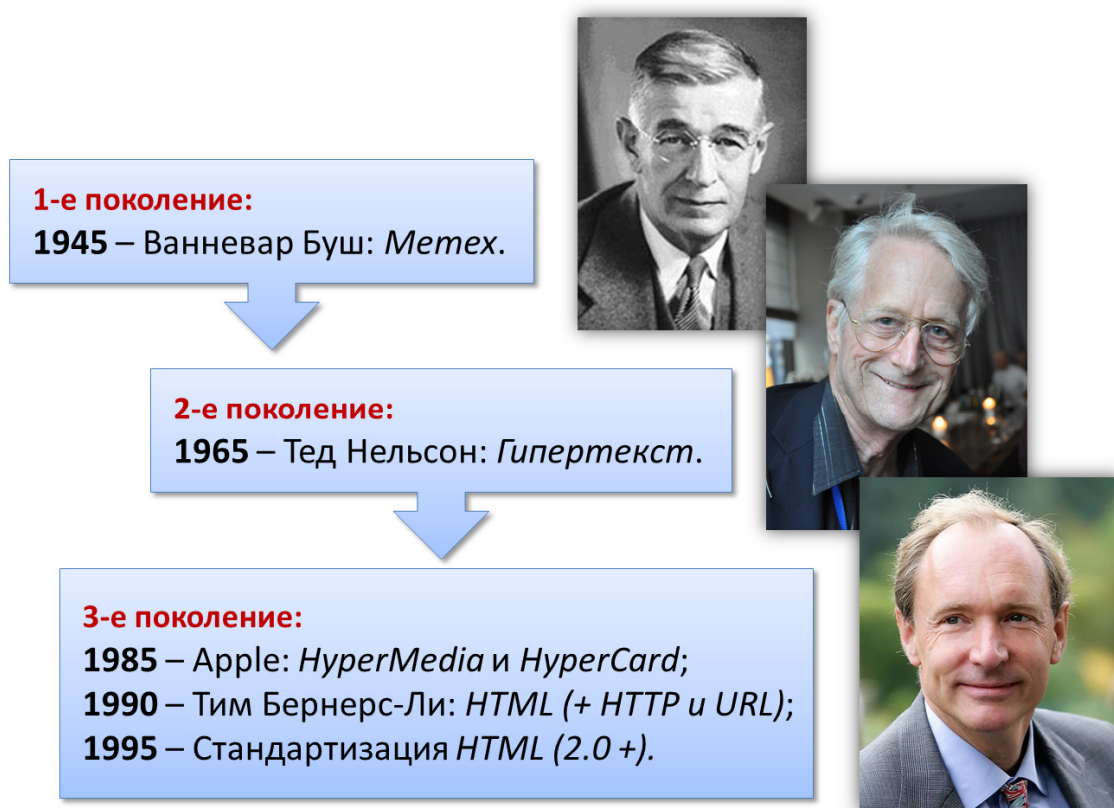


Рис. 2. Три поколения развития гипертекста и ключевые ученые-разработчики (вверху (1-е поколение) – Ванневар Буш; по середине (2-е поколение) – Тед Нельсон; внизу (3-е поколение) – Тим Бернерс Ли)

Первое поколение развития началось с формирования идеи гипертекста, которая была выдвинута *Ванневаром Бушем* (англ. *Vannevar Bush*) в 1945 году в предложениях по созданию **электромеханической информационной системы MEMEX**.

В 1940 году американский инженер и разработчик аналоговых компьютеров Ванневар Буш был назначен председателем Государственного комитета оборонных исследований США и советником президента по науке при президенте Рузвельте. В 1944 году Рузвельт запрашивает у Буша рекомендации о том, какие уроки Второй мировой войны следует извлечь США, на что по-

лучает ответ: «...государственные интересы в области науки и образования могут быть наилучшим образом достигнуты созданием Национального фонда науки».

В период 1941–1947 гг. Ванневар Буш возглавлял бюро научных исследований и разработок при правительстве США и являлся председателем Комитета по военной политике, занимаясь координацией усилий научного сообщества (6000 ведущих ученых страны) в целях военной обороны, разработкой ядерного оружия и *Манхэттенским проектом*. В 1950 г. В. Буш возглавил созданный к тому времени *Национальный фонд науки* (NSF).

В 1945 году Ванневар Буш в нашумевшей статье «Как мы можем мыслить» (англ. *As We May Think*) предлагает прообраз гипертекстового устройства Memex: «Человек построил столь сложную цивилизацию, что он нуждается в механизмах обработки данных, которые уже не вмещаются в его ограниченную память...».

Устройство *MEMEX* (англ. *MEMory EXtender* — «расширитель памяти») в представлении Буша призвано помочь человеку хранить все его книги, все его записи и все его коммуникации с другими людьми (рис. 3). Устройство выглядит как обычный стол, на котором клавиатура, кнопки и рычажки. Небольшая часть стола занята данными в виде микрофильмов, остальная часть — рабочий механизм.

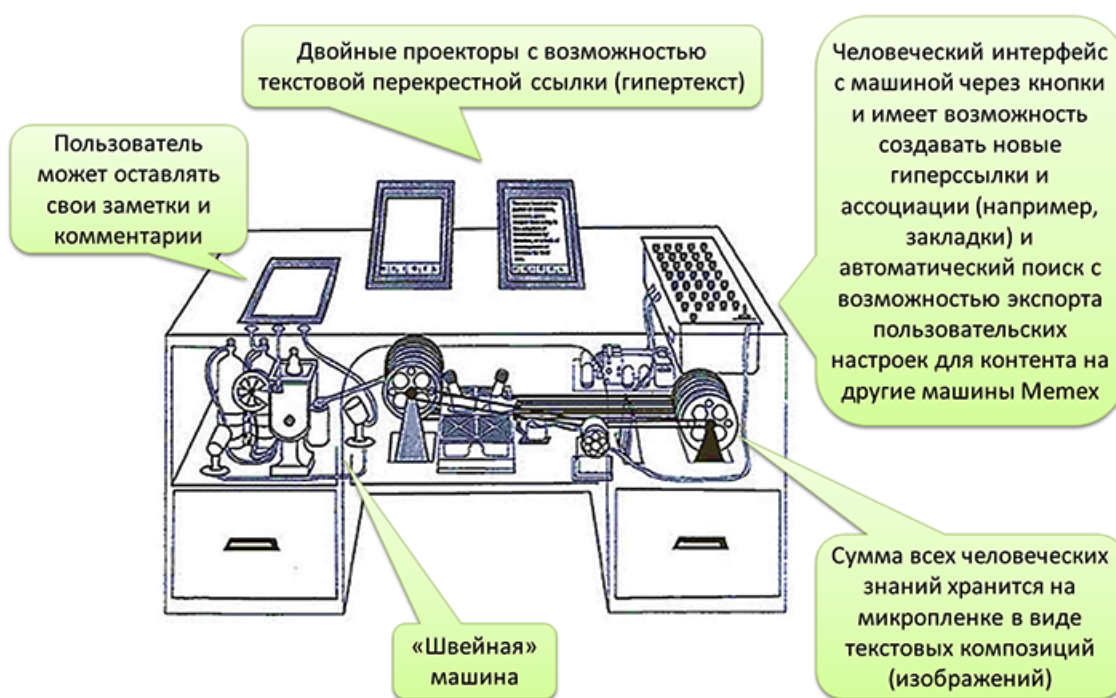


Рис. 3. Концепция Memex в представлении Буша

Книги всех типов, картинки, газеты могут быть немедленно получены и включены в систему. Эта система использует индексирование — если человек хочет получить доступ к книге — он набирает необходимый код на клавиатуре и нужная книга или страница возникает перед ним на экране MEMEX.

Когда пользователь строит ассоциативную цепочку между двумя документами, то он записывает название цепочки в книгу кодов. Сохраненные цепочки могут быть доступны пользователю в любое время. Они образуют совершенно новую книгу, которая хранится внутри MEMEX и может быть вызвана из его памяти и через много лет.

Несмотря на то, что Буш был советником по науке президента Рузвельта, его призывы к физической реализации электро-механической информационной системы оказались безуспешными — идея создания Memex так и не была реализована.

С 1965 года начинается **второе поколение** развития гипертекста. Именно тогда американским философом, социологом (по первому образованию) и первооткрывателем в области информационных технологий *Тедом Нельсоном* (англ. *Ted Nelson*) был **введен в научное обращение термин «гипертекст»** для описания документов, которые выражают нелинейную структуру идей в противоположность линейной структуре традиционных книг, фильмов и речи.

В период 1960–1965 гг. Нельсон занимался разработкой текстового редактора на низкоуровневом языке ассемблера (англ. *assembly language*). В докладе о редакторе на ежегодной конференции «Association of Computing Machinery» (1965 г.) Тед Нельсон определил понятие «гипертекст» следующим образом: *«Гипертекст — это непоследовательный способ записи, текст ветвится, и это позволяет читателю осуществлять выбор и перемещаться по нему с помощью интерактивного экрана»*.

Более поздний термин «гипермедиа» близок к определению Нельсона по смыслу, но он подчеркивает наличие в гипертексте нетекстовых компонентов, таких как анимация, записанный звук и видео.

В начале 60-х годов XX в. группа программистов во главе с Тедом Нельсоном начала работу над **проектом «Xanadu»**, в котором воплощалась идея гипертекста. Нужно понимать, что эта идея появилась задолго до того, как была реализована первая си-

стема гипертекстовых связей в документах. Модель работы Xanadu (рис. 4) определяла следующее: каждый документ имеет оригинальную форму (виртуальный файл), но поддерживает видоизменение с версионностью, редактирование и бесконечное дополнение с индексированием.

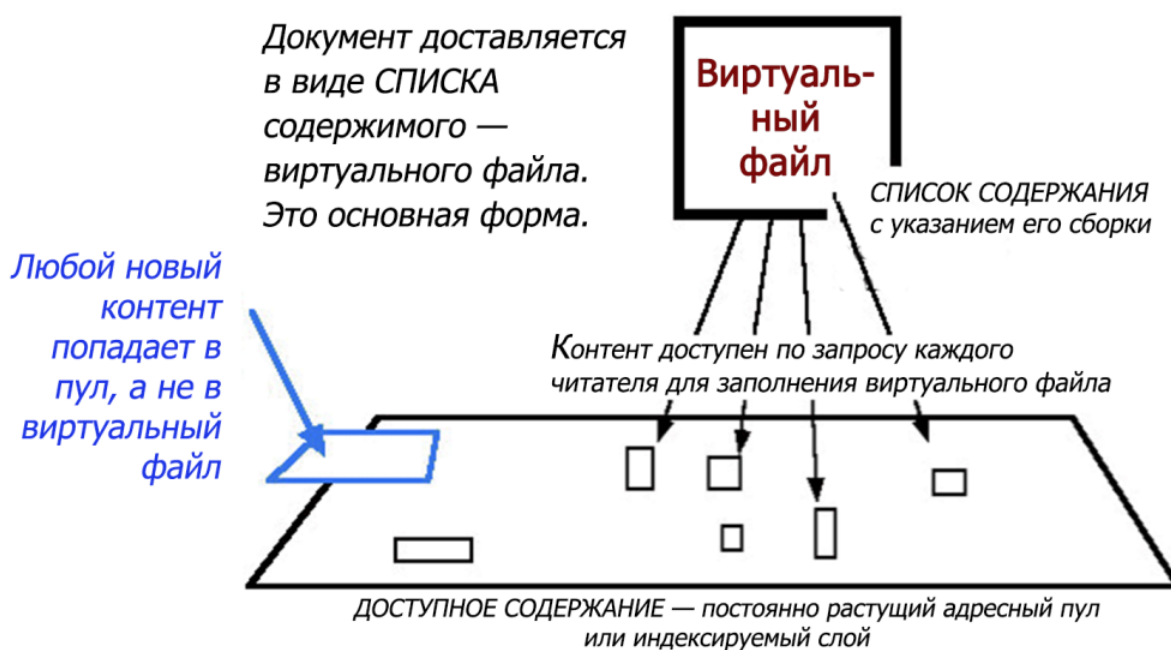


Рис. 4. Модель документа Xanadu построена на предположении о постоянном изменении и повторном использовании

Проект Xanadu начался с появлением интерактивных экранов — с предвидения новой «экранной литературы» из параллельных, взаимосвязанных документов. Таким образом, оказывается возможным показать точную связь между произведениями на разных страницах. Видимая связь становится структурной частью письма, такой же фундаментальной, как абзац или заголовок. Это обеспечивает обобщенный способ представления параллельных документов и их точных связей.

Также ко второму поколению развития гипертекста относится целый ряд исследовательских коллективов, занимающихся гипертекстом. Одна из наиболее успешных групп существовала при американском Университете Брауна (англ. *Brown University*, штат *Род-Айленд*) в научно-исследовательском институте информационной и преподавательской деятельности (*Institute for Research in Information and Scholarship*, IRIS). В рамках этой группы был создан проект **Intermedia**, руководителем и главным

архитектором которого был содиректор института *Норман Мейровиц (Norman Meyrowitz)*. Проект был рассчитан на двадцатилетнюю работу и опирался на опыт, извлеченный из разработки предшествующих гипертекстовых систем:

1) **Гипертекстовая система редактирования Hypertext Editing System (HES)**, разработанная в 1967 году *Тедом Нельсоном* и несколькими студентами Брауновского Университета (рис. 5). Хьюстонский Центр космических полетов с человеком на борту использовал эту систему для подготовки документации по программе «Аполлон».



Рис. 5. HES-консоль, университет Брауна, 1969 г.

2) **Система поиска записей и редактирования File Retrieval and Editing System (FRESS)**, разработанная в 1968 году в Университете Брауна американским профессором информатики *Андрисом ван Дамом (Andries van Dam)* и его студентами (уже без участия Теда Нельсона). В начале 70-х годов концерном *Philips* был сделан ее коммерческий вариант. Более 10 лет сотни преподавателей и студентов использовали эту систему для обучения (например, учебный класс английской поэзии вел чтение и записи целиком на гипертекстовом документе, находившемся в общем пользовании). Система FRESS была многопользовательской,

предусматривала динамическую иерархию и двунаправленные связи, а также узлы и связи с выделением ключевых слов. На графических терминалах поддерживались многооконные изображения и векторная графика.

3) **Электронная документационная система Electronic Document System (EDS)** — гипертекстовая система с акцентом на цветную растровую графику и механизмы навигации (рис. 6). EDS — это исследовательский проект американских ученых *Стивена Фейнера (Steven K. Feiner)*, *Сандора Нагу (Sandor Nagy)* и *Андриса ван Дама* в 1978–1981 гг. в Университете Брауна.



Рис. 6. Система EDS – концепция электронного руководства по техническому обслуживанию (1980 г.). Программное обеспечение фактически работало на специальном растровом дисплее VAX 11/780

Система Intermedia — часть всеобъемлющих усилий, предпринимавшихся в Брауновском Университете с целью повышения эффективности применения рабочих станций в учебных аудиториях. В этом контексте она разрабатывалась как некоторый комплекс инструментальных средств, позволяющих авторам-пользователям устанавливать связи с текстами, динамическими рядами (*timelines*), диаграммами и другими изображениями, гене-

рируемыми компьютером, а также с документальными видеофильмами и музыкальными записями. Система поддерживала двунаправленные связи для текста и графики. Маленькие иконки использовались для маркировки точек их прикрепления (якорьков — anchors). Связи хранились отдельно от содержимого базы данных.

Система Intermedia разрабатывалась и как средство, позволяющее преподавателям Университета Брауна организовывать и представлять свой учебный материал на компьютере, и как интерактивная среда для студентов, работая в которой они могли изучать этот материал, а также добавлять к нему свой собственный в виде аннотаций или сообщений.

В 1975 году идея гипертекста нашла воплощение в **информационной системе внутреннего распорядка** в качестве ключевого интерфейса между пользователями и материально-техническим обеспечением *атомного авианосца «Карл Винстон» (USS Carl Vinson, CVN-70)*, которая получила название **ZOG**. ZOG была гипертекстовой системой, ранее разработанной в американском Университете Карнеги-Меллона (*Carnegie Mellon University (CMU), Питтсбург, штат Пенсильвания*) в 1970-х годах *Дональдом Маккракеном (Donald McCracken)* и *Робертом Аксцином (Robert Akscyn)*, чтобы служить в качестве интерфейса для программ искусственного интеллекта и когнитивных наук, собранных в CMU на летнем семинаре.

ZOG состоял из «фреймов», содержащих заголовки, описание, строку, содержащую системные команды ZOG, и выбор (пункты меню), которые вели к другим фреймам. ZOG впервые применил «рамочную» или «карточную» модель гипертекста, позже популяризированную в HyperCard фирмы Apple. В таких системах рамки или карточки не могут прокручиваться для отображения содержимого, которое является частью того же документа, но находится за пределами экрана. Вместо этого текст, размер которого превышает размер одного экрана, должен быть помещен на другой (который затем составляет отдельный фрейм или карточку).

Позже проект ZOG стал коммерческим продуктом для долгосрочных исследований искусственного интеллекта, проводимых Алленом Ньюэллом (*Allen Newell*) и финансируемых Управлением военно-морских исследований. В коммерческом варианте эта

система получила название «Система управления знаниями» — KMS (*Knowledge Management System*).

В третьем поколении развития гипертекста появлялись реализации типа HyperCard компании Apple или NodeCards компании Xerox.

Один из ключевых разработчиков программного обеспечения для компьютеров Apple Macintosh, *Билл Аткинсон (Bill Atkinson)*, вдохновленный идеями Ванневары Буша и его проектом Xanadu, внес большой вклад в популяризацию гипертекста, создав в 1987 году программу **HyperCard**. Это, по сути, визуальная среда программирования, позволяющая создавать собственные приложения гипермедиа на компьютере Macintosh в системе Mac OS (рис. 7). С точки зрения пользователя, HyperCard выполняет функцию «организатора информации», позволяющего собирать воедино тексты, картинки, звуки, анимацию и предлагая их пользователю в удобном интерактивном виде.

Основной информационной единицей в HyperCard был компьютерный аналог карточки хранения информации в базе данных. Несколько карточек, объединенных по какому-то принципу, составляют так называемый *информационный стек* (от англ. *stack* — куча, груда, хранилище), который может ассоциироваться с гипертекстовым документом или книгой. При этом принципам гипертекста в HyperCard обладает возможность создания в любом месте информационной карточки специального значка (иконки), отсылающего к тексту другой карточки. Стеки HyperCard содержат различную информацию (изображение, текст, элемент интерфейса, звук, видео, анимация и пр.).

В HyperCard использовался **язык сценариев HyperTalk**, с помощью которого пользователь мог создавать как сами связи между карточками, так и огромное число дополнительных функций.

Сам принцип хранения информации в виде стека требовал, чтобы на каждой карточке присутствовали как минимум *две функциональные кнопки-иконки* — перехода к следующей карточке и возврата назад, к предыдущей карточке. Простота и легкость формирования связей между карточками в системе HyperCard способствовали тому, что иконку-ссылку можно было установить в любом месте карточки и связать эту карточку с любой другой в соответствии с намерением и желанием пользователя.

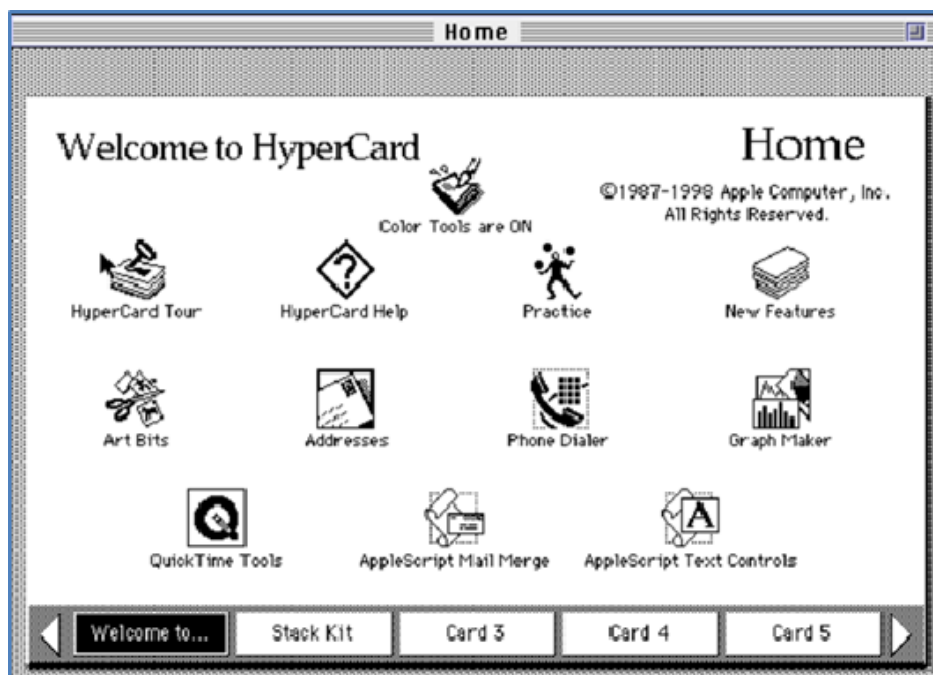


Рис. 7. Домашняя страница HyperCard

Программа HyperCard была включена в системное ПО Macintosh и требовала всего 1 Мб оперативной памяти. HyperCard добилась успеха практически мгновенно — в первый год количество проданных копий превысило 1 млн. Окончательный релиз HyperCard 2.4.1 был выпущен в 1998 году.

В 1983 г. увидела свет написанная на языке программирования Lisp программа ведения заметок с применением перекрестных ссылок **NoteCards**, разработанная компанией Xerox. NoteCard обладала «прокручиваемыми» окнами, возможностью работы с единой системой форматирования для всех заметок, а также *многооконной системой*, в которой просмотрщику заметок (прообраз будущих браузеров) было выделено собственное окно.

В 1987 г. была проведена первая **специализированная конференция Hypertext'87**, материалам которой был посвящен специальный выпуск журнала «Communication ACM».

К 1989 году гипертекст представлял новую, многообещающую технологию, которая имела относительно большое число реализаций с одной стороны, а с другой стороны делались попытки построить формальные модели гипертекстовых систем, которые носили скорее описательный характер и были навеяны успехом реляционного подхода описания данных.

4. Браузеры. История развития

После включения компьютера, который подключен к Интернету, нельзя получить какие-либо данные из сети. Для этого нужно использовать специальное программное обеспечение, которое будет делать запрос на информацию, передавать/получать ее, редактировать, и отображать на мониторе пользователя. Этим и занимается браузер, которому можно дать следующее определение:

Веб-обозреватель, интернет-обозреватель, браузер (от англ. *Web browser*; *browse* — просматривать, листать) — программное обеспечение для навигации и просмотра содержимого веб-ресурсов, то есть для запроса веб-страниц (преимущественно из сети Интернет), их обработки, вывода и перехода между страницами. С помощью браузера также можно просматривать и загружать различную текстовую, графическую, звуковую и видеoinформацию.

Первый браузер появился более двадцати лет назад. За эти десятилетия произошла трансформация браузеров из «примитивных» приложений, в довольно мощные и многофункциональные программы для работы в Интернете. Мы полагаем, что вам, как активным пользователям браузеров, стоит знать, как это было!

Наша жизнь тесно сплетена со многими окружающими нас вещами, настолько тесно, что мы иногда перестаем их замечать. Однако каждое творение человека имеет собственную историю, наполненную яркими событиями, фактами и даже случайными стечениями обстоятельств. История браузеров также насыщена яркими моментами, и позволяет больше узнать как о развитии программного обеспечения (ПО) для веб-ресурсов, так и о развитии сети Интернет в целом. Ведь современные браузеры остаются обыкновенными компьютерными программами лишь с формальной стороны, однако фактически они выполняют функцию связующего звена между Интернетом и человеком, и от того, как они справляются с возложенными на них задачами, зависит и наше восприятие виртуального мира.

Историческим началом развития браузеров считается следующее событие: автор концепции Web и создатель первого в мире сайта Тим Бернерс-Ли выпустил текстовый браузер под

названием «**WorldWideWeb**» (рис. 8). Его разработка пришлась на октябрь-декабрь 1990 года. Браузер работал в популярной в то время операционной системе NextStep, разрабатываемой компанией Next Стива Джобса. Идея Тима Бернса-Ли была подхвачена многими компаниями, которые начали разрабатывать свои собственные текстовые браузеры. Однако заметную «видимость» браузеры приобрели с момента появления у них **графического интерфейса**, что также сказалось и на широком распространении Интернета.

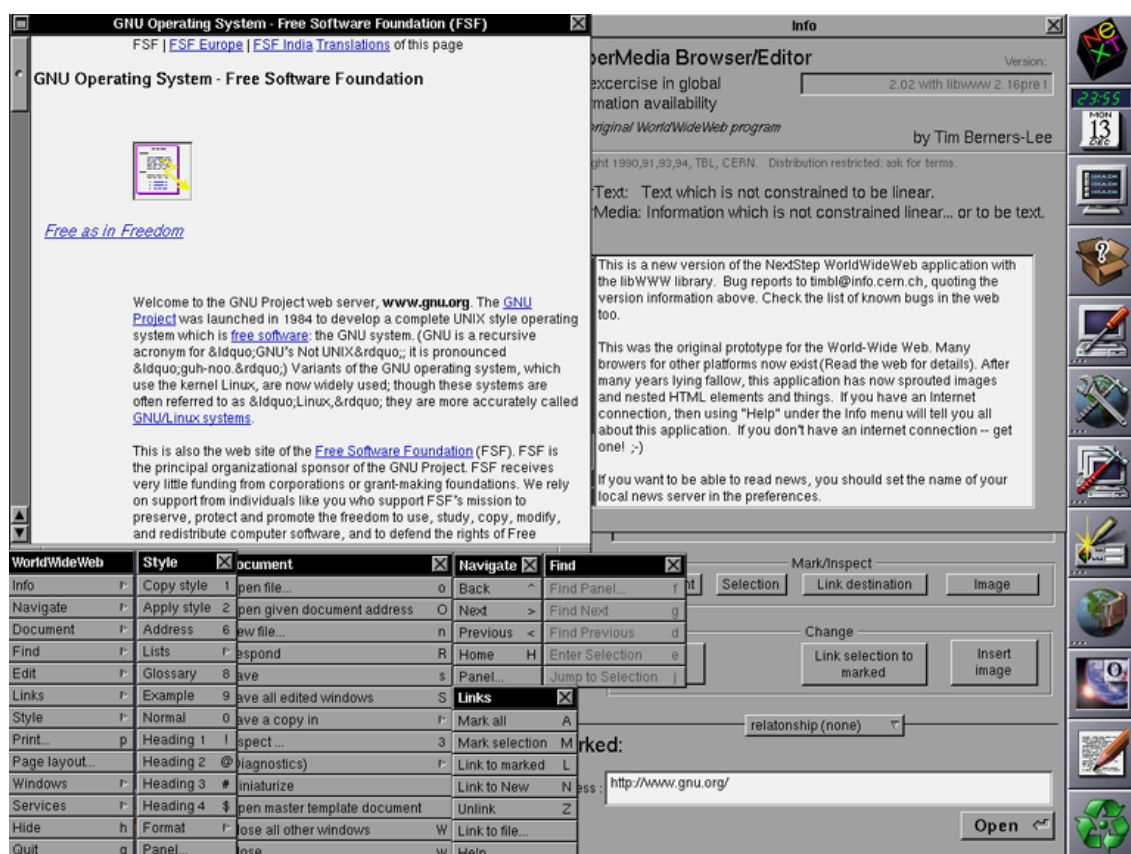


Рис. 8. Браузер WorldWideWeb с демонстрацией многих из его функций

Первым распространенным браузером с графическим интерфейсом стал **NCSA Mosaic** (рис. 9), который впервые появился 22 апреля 1993 года. Работа по развитию браузера Mosaic была прекращена в 1997 году, однако авторы открыли исходные коды программы, что позволило разработчикам компаний Netscape и Microsoft создать свои браузеры на основе данных исходных кодов. Браузер NCSA Mosaic имел свои недостатки, но почти все они были устранены в браузере **Netscape Navigator** (рис. 10). Компания Netscape (некоторые сотрудники компании Netscape

были из NCSA и участвовали в разработке Mosaic) выпустила Netscape Navigator под разные операционные системы (UNIX, Windows, Mac OS) и добилась заметного успеха, в том числе и коммерческого. Это побудило компанию Microsoft выпустить свой браузер **Internet Explorer (IE)** (рис. 11). Netscape Navigator был самым широко используемым браузером и Microsoft приобрела лицензию на Mosaic, чтобы создать Internet Explorer 1.0, выпущенный как часть пакета Windows 95 Plus! в августе 1995 года. Спустя три месяца для бесплатной загрузки был выложен браузер Internet Explorer 2.0. В отличие от Netscape Navigator (Netscape Navigator был бесплатным для домашних пользователей, но корпорациям приходилось покупать его за \$ 99), он был бесплатно доступен всем пользователям Windows, даже коммерческим компаниям. Другие компании последовали примеру Microsoft и сделали свои браузеры бесплатными.

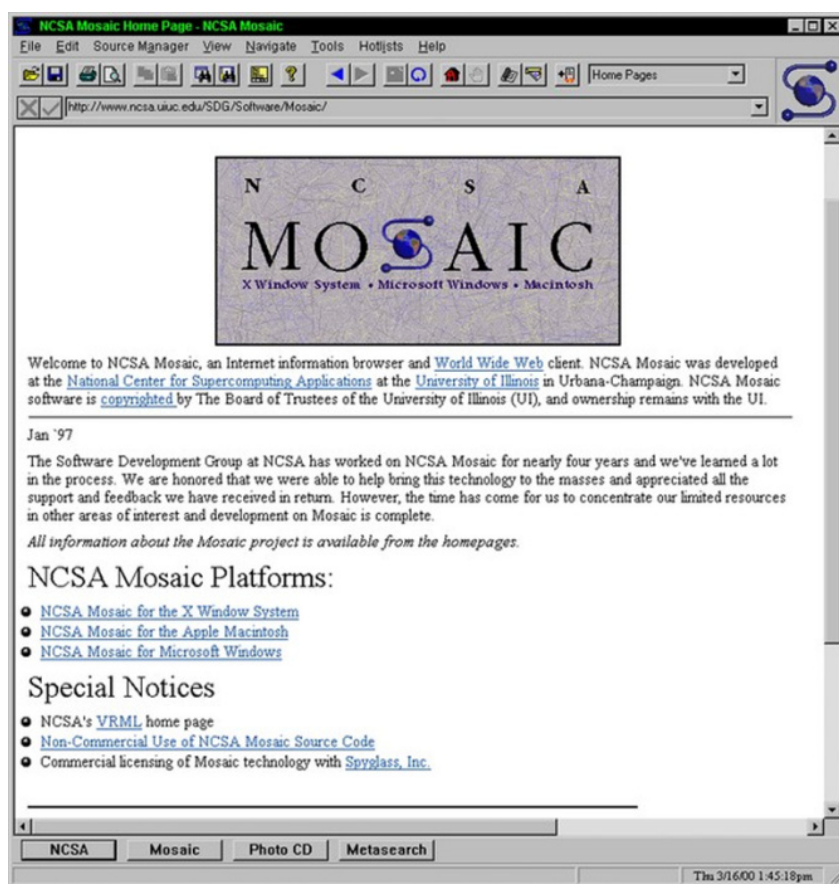


Рис. 9. Браузер NCSA Mosaic версии 3.0

Какое-то время браузеры конкурировали честно, пока все падающая доля Internet Explorer не заставила Microsoft прибегнуть к иным методам завоевания рынка.



Рис. 10. Браузер Netscape Navigator версии 4.04

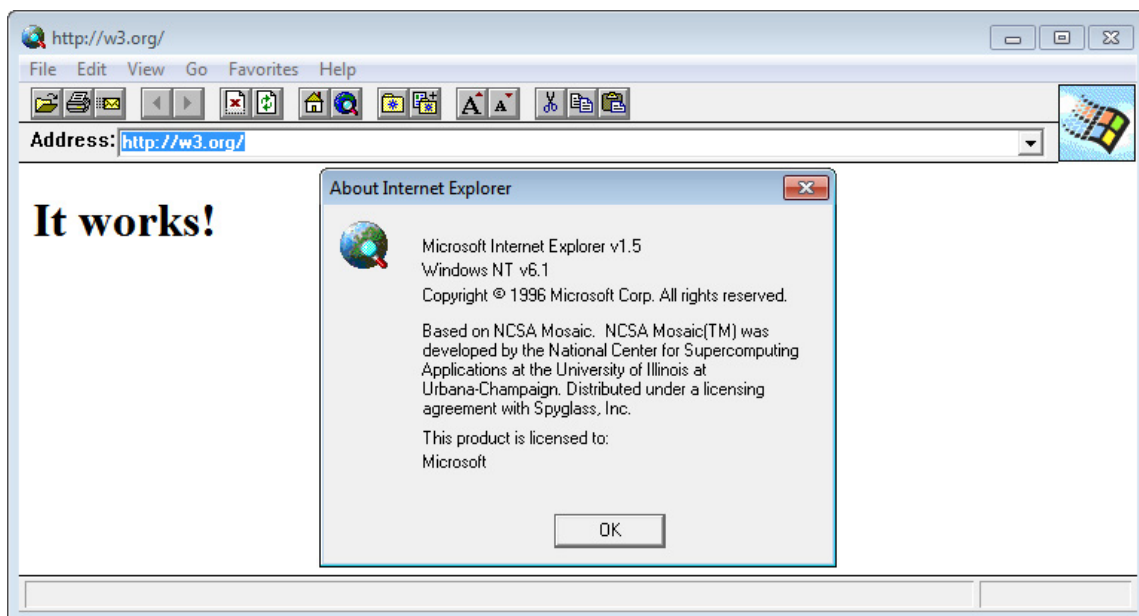


Рис. 11. Одна из первых версий браузера Internet Explorer

Речь идет о включении Internet Explorer в операционную систему Windows 95. Сразу после выпуска в ОС Windows 95 Internet Explorer не был встроен браузер, но он появился позднее, вместе с обновлением, получившим название Windows 95 OSR2. Это

был Internet Explorer 3.0. Появление браузера, встроенного операционную систему значительно изменило расклад сил. Netscape Navigator стал стремительно терять популярность, поскольку пользователи Windows не искали другого браузера, а пользовались встроенным. Поэтому, вместе с ростом популярности Windows, росла популярность Internet Explorer.

Есть и еще один фактор популярности браузера. Когда IE только появился, он не поддерживал фреймы, а потому вебмастера, создающие сайты, настраивали их таким образом, что при получении User-Agent Internet Explorer, сервер отдавал страницу без фреймов. Потом IE научился работе с фреймами, но вебмастера не спешили перенастраивать сервера. IE пришлось маскироваться под Netscape Navigator, сообщая, что он совсем не IE, а Mozilla.

Почему «Mozilla»? Дело в том, что Mozilla — это внутреннее имя браузера Netscape Navigator, обозначающее Mosaic Killer (убийца Mosaic). Название показалось сотрудникам фирмы чрезмерно вызывающим, поэтому не стало официальным. Впрочем, позднее название возродилось в потомке Navigator.

Internet Explorer сделал свое «черное дело». Став монополистом рынка, этот браузер заставил всех вебмастеров создавать страницы специально под него. При этом в Microsoft не слишком заботились о соблюдении веб-стандартов, что привело к ситуации: сайты создавались не по стандартам, а по тому, как решили реализовать ту или иную функцию в Microsoft.

Середину 1995 года можно считать началом **первой войны браузеров (1995–1999)**, закончившейся полным падением Netscape и триумфом Internet Explorer, занявшего более 95 % рынка (рис. 12). В таком состоянии браузер Internet Explorer дожил до своего апогея — Internet Explorer 6, вышедшего за несколько недель до выпуска Windows XP. Но тут началась **вторая браузерная война (2004 — настоящее время)**.

Теряющая рынок компания Netscape была куплена корпорацией AOL, которая продолжила традицию NCSA, выпустившей исходные коды Mosaic в свободный доступ, и поступила также с кодами Navigator. Это действие и оказалось той самой «бомбой замедленного действия», которая впоследствии разрушила монополию Microsoft.

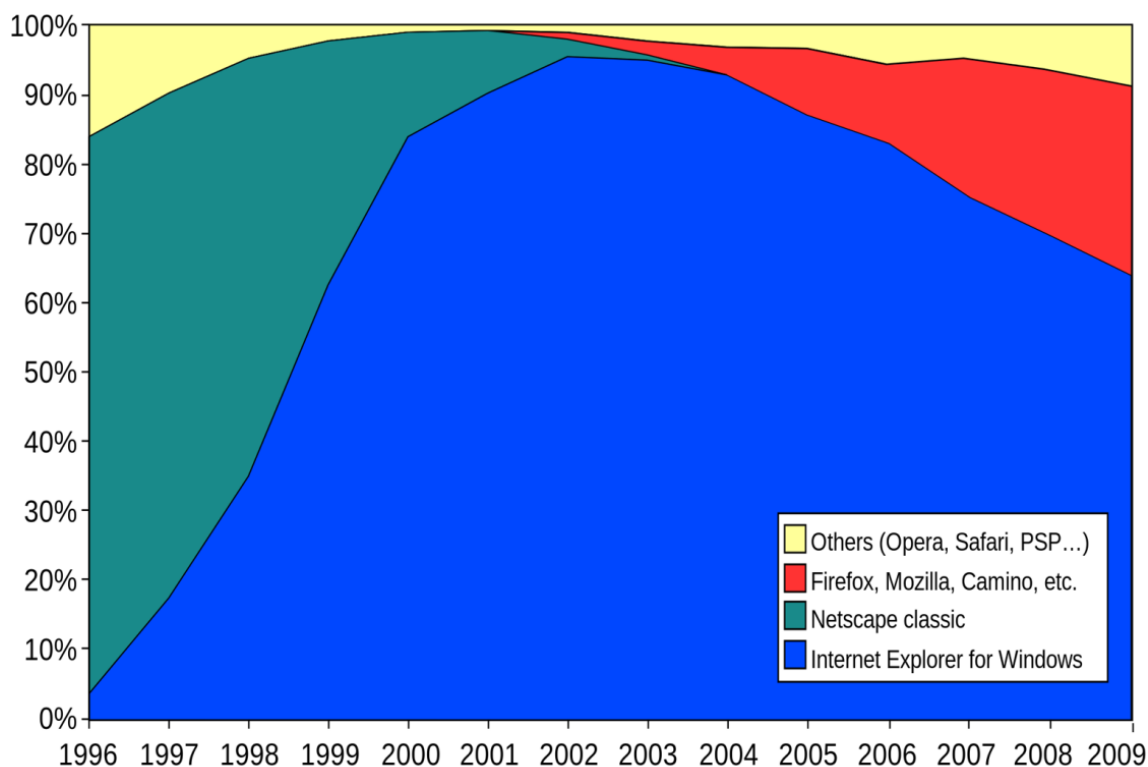


Рис. 12. Браузерные войны: конец первой и начало второй

В 2000 году был выпущен Netscape Navigator 6, представляющий собой браузер с совершенно новым движком¹ **Gecko**, который с самого момента своего создания был направлен на полную поддержку общепринятых веб-стандартов. Netscape Navigator 6 основывался на разработках проекта Mozilla Application Suite, многофункционального приложения, включающего в себя кроме собственно браузера, еще и почтовый клиент, календарь, IRC-клиент, HTML-редактор и средства веб-разработки.

Популярность Mozilla Suite оказалась выше, чем у основанной на ней Netscape Navigator, так что спустя несколько лет, AOL отказалась от развития браузера. Исходные коды были переданы проекту Mozilla Foundation.

Mozilla Foundation приняла решение о разделении Mozilla Suite на составляющие части. Получившийся браузер получил название «Phoenix», в честь птицы Феникс, сгорающей и возрождающейся.

¹ **Браузерный движок** (англ. layout engine) — программа, преобразующая содержимое веб-страниц (файлы HTML, XML, цифровые изображения и т. д.) и информацию о форматировании (CSS, XSL и т. д.) в интерактивное изображение форматированного содержимого на экране.

дающей из пепла. Таким образом, разработчики хотели показать преемственность Phoenix по отношению к Netscape Navigator. Вскоре он был переименован на «Firebird» («Жар-птица»), а затем в Firefox («Огненная лиса»). Переименования связаны с тем, что два предыдущих названия уже использовались в качестве наименований программных продуктов у других производителей.

На фоне «угасающего» Netscape Navigator, в 1996 году появился браузер **Opera** от норвежской компании с одноименным названием. Многие пользователи в России начали использовать этот браузер, потому что он был быстрее и легче Internet Explorer. Но популярность данного браузера долгое время сдерживала методика его распространения. Дело в том, что с самого своего появления Opera предлагалась своим пользователям по условно-бесплатной модели: в правом верхнем углу браузера находился рекламный блок, а если пользователь хотел его удалить, требовалось заплатить за это. Естественно, что при наличии бесплатных альтернатив мало кто желал использовать браузер с рекламой.

В 2005 году норвежские разработчики браузера поняли бесперспективность использования условно-бесплатной модели и сделали браузер полностью бесплатным.

В 2006 году на рынке браузеров для Windows появился еще один крупный игрок: компания Apple выпустила для ОС Windows свой браузер под названием **Safari**. Safari под Windows, так же как и его версия под Mac OS X работал на революционном движке **WebKit**. В Apple значительно доработали движок и, в соответствии с лицензией, предоставили исходники всем желающим.

Необычность внешнего вида, современные технологии, находки в интерфейсе, бренд Apple — все это позволило браузеру обрести значительное количество поклонников. Вместе с тем, Safari для Windows не поддерживал расширения, что сразу отнимало у него большое количество очков.

Спустя два года, в 2008 году, компания Google совершенно неожиданно для всех решила выйти на рынок браузеров. Она выпустила кроссплатформенный браузер с открытым исходным кодом, под названием **Chrome**. В браузере использовался уже знакомый нам движок WebKit. Google сообщила, что покажет поль-

зователям, каким именно должен быть идеальный браузер. В Google Chrome, впервые в истории браузеров, была использована технология изолированных процессов. Кроме того, пользователю впервые предлагался режим «инкогнито», позволяющий не оставлять никаких «следов» в браузере.

И, наконец, в январе 2012 года на технологической конференции Yet another Conference был представлен браузер **Яндекс.Браузер**, созданный компанией «Яндекс». Разработчики постарались использовать не только все лучшее от других браузеров, и добавить свои «фишки». Так, главной особенностью данного браузера стала его интеграция с сервисами самого Яндекса: почтой, переводчиком, «Яндекс.Диском», поиском «Яндекс» и пр.

В наши дни пользователям доступно большое количество самых разнообразных браузеров, но наиболее популярными являются **Google Chrome**, **Internet Explorer**, **Mozilla Firefox**, **Apple Safari** и **Opera**. Каждый из пятерки обладает неким базовым функционалом, который способен удовлетворить нетребовательного пользователя. В тоже время любой браузер имеет уникальные функции, присущие именно ему. Эти уникальные функции и являются причиной, по которой пользователи выбирают тот или иной браузер.

В настоящее время монополия Internet Explorer полностью окончена. Это пошло на пользу не только Вебу, который стал развиваться независимо от браузеров и строится по общепринятым стандартам, но и, в конечном счете, самому Internet Explorer. Возросшая конкуренция заставила программистов Microsoft потрудиться над новыми версиями браузера. И если до сих пор он по ряду параметров и не догоняет своих конкурентов, то уж, во всяком случае, ушел далеко от антипатичного Internet Explorer 6.

Перед каждым пользователем рано или поздно встает вопрос выбора браузера, ведь браузер является одной из тех программ, которой люди, ежедневно бороздящие просторы Интернета, пользуются чаще всего. Преподаватели курса «Интернет-технологии», — в силу его специфики, пользуются одновременно всеми распространенными браузерами.

В браузерах разных производителей есть очень много общего. Первое, что бросается в глаза, это минимизация элементов управления — все место отдано под окно отображения веб-страниц. Далее — теперь все браузеры имеют «вкладочный» интерфейс (к слову, так было не всегда) — каждая страница может открываться в новой вкладке. Каждый браузер прост в использовании и после установки сразу готов к работе, не требуя особых настроек.

Также в каждом браузере есть возможность изменения его внешнего вида и поведения (как правило, это результат манипуляций расширенными настройками). Каждый из них поддерживает современные технологии HTML и CSS, отображает графику, воспроизводит видео и звук. Кроме того, все они способны сохранять веб-страницы, создавать закладки, вести историю и загружать файлы по FTP- или HTTP-протоколам передачи данных с возможностью возобновления загрузки («докачки») при сбоях. Некоторые браузеры могут открывать doc- и pdf-документы, хранить пароли, создавать многопользовательские структуры, синхронизировать пользовательские настройки и работать с облачными с облачными технологиями и сервисами.

Пробуйте, экспериментируйте, и вы быстро найдете для себя оптимальный браузер!

5. Развитие языка гипертекстовой разметки HTML

В 1969 году сотрудник компании IBM *Чарльз Голдфарб* (*Charles Goldfarb*) возглавил проектирование компьютерной системы обслуживания юридических контор. Под его руководством был создан **первый язык разметки документов Generalized Markup Language (GML)**, в котором была реализована концепция типа документа (формально определенного шаблона, описывающего схему внутреннего построения схожих документов) и вложенных друг в друга структур. GML не зависел ни от марки компьютеров, ни от операционной системы, и IBM удалось перевести 90 % своей документации в этот формат.

К 1978 г. *комитет по обработке информации Американского национального института стандартов* (англ. *American national standards institute, ANSI*) всерьез заинтересовался языками подготовки гипертекстовых данных. Чарльз Голдфарб возглавил комитет по новому направлению, связанному с формированием стандарта для мощного метаязыка разметки документов, который был назван **стандартным обобщенным языком разметки SGML** (англ. *Standard General Markup Language*), в основу которого был заложен GML. Первый рабочий вариант спецификации появился в 1980 году, а в 1983 г. Ассоциация GCA (англ. *Global Citizens Association*) приняла шестую рабочую версию SGML в качестве промышленного стандарта (GCA 101-1983), поддержанного Министерством обороны и Налоговым управлением США. Спустя два года, в 1985 г., сформировалась международная группа пользователей SGML. Считается, что начиная с этого года, человечество вступило в новую эру — развитие современных гипертекстовых технологий.

В 1986 г. международная организация по стандартизации ISO (англ. *International Organization for Standardization*) одобрила **стандарт SGML (ISO-8879)**. Он позволил отказаться от конкретных способов представления информации и сосредоточить усилия на продумывании структуры документов с помощью правил определения собственных тегов форматирования, их атрибутов и синтаксиса использования.

SGML оказался очень мощным и универсальным языком, так как требовал точного описания всех нюансов создаваемого

синтаксиса документа и подробных правил формирования тегов. В рамках SGML была разработана **концепция DTD** (англ. *Document Type Definition* — определение типа документа) которая позволила связать конкретные *синтаксические правила разбора с заданными способами организации структуры документов*. Взяв на вооружение новую концепцию DTD, многие компании приступили к активной разработке программ анализа SGML-текстов.

Наступил 1989 год, который вошел в историю как **год появления идеи Всемирной паутины** (англ. *World Wide Web, WWW*)! Эта идея была предложена (и реализована всего через два года) **Тимоти Джоном Бернерсом-Ли** (*Timothy John «Tim» Berners-Lee*) (рис. 13), программистом *Европейского центра ядерных исследований* (фр. *Conseil Européen pour la Recherche Nucléaire, CERN*) — крупнейшей в мире лабораторией физики высоких энергий, расположенной в Женеве.



Рис. 13. Сэр Тимоти Джон Бернерс-Ли (1992 г.). Создатель URI, URL, HTTP, HTML и Всемирной паутины. Действующий глава Консорциума Всемирной паутины. Автор концепции семантической паутины и множества других разработок в области информационных технологий

Родители Тима Бернерса-Ли были математиками и участвовали в создании «Manchester Mark I» — одного из первых компьютеров. Обучаясь в Королевском колледже в Оксфорде, Тимоти самостоятельно собрал свой первый компьютер на базе про-

цессора M6800. Работа в CERN также была отмечена группой проектов, в которых Тим принимал активное участие: создание программы «Enquire» — «Дознаватель» (1980 г.), разработка распределенных систем для сбора научных данных (1984 г.), разработка внутренней системы обмена документов для ENQUIRE (1989 г.). В результате работы над системой обмена документов был предложен глобальный гипертекстовый проект — новое SGML-приложение, известное сегодня как **язык гипертекстовой разметки HTML**.

В 1990 году Тим Бернерс-Ли написал на языке программирования Objective-C **первый браузер для компьютера NeXT** (рис. 14). Браузер включал в себя также первый редактор HTML-документов. Не отставало и наше государство (тогда еще не отставало...) — в декабре 1990 г. при Министерстве связи СССР был открыт **Научно-технический центр гиперинформационных технологий**, известный как ГНТЦ «Гинтех».



Рис. 14. Исторический компьютер NeXT, которым пользовался Тим Бернерс-Ли в 1990 году (в настоящее время экспонат выставки Microsoft в CERN). Это был первый веб-сервер «httpd», гипермедиа-браузер и веб-редактор. Поврежденная этикетка на кубическом системном блоке гласит: «Эта машина является сервером – НЕ ВЫКЛЮЧАЙТЕ ЕГО!!»

Бернерс-Ли начал с философии, согласно которой большая часть академической информации должна быть бесплатно доступна для всех. Он разрабатывал Всемирную паутину как средство для обеспечения доступа к информации для разрозненных по всему миру академических групп через сеть Интернет. Он объединил методы поиска информации и гипертекста, чтобы создать простую, но мощную глобальную информационную систему.

Первый в мире сайт был запущен **6 августа 1991 года** по адресу <http://info.cern.ch> (он и сейчас доступен в Сети). На сайте описывается, что такое Всемирная паутина, как установить веб-сервер, как получить и установить свободно распространяемые браузер, веб-редактор и т. п. Этот сайт являлся также первым в мире интернет-каталогом.

В то же время Тим Бернерс-Ли, работая на компьютере NeXT, завершил **первую версию гипертекстового браузера «WorldWideWeb»**. Эта программа является предшественником большей части того, что мы сегодня называем или знаем как «Интернет». В то время как Интернет в основном рассматривается как среда для потребления информации, браузер WorldWideWeb позволял пользователю редактировать и создавать веб-страницы. Это отражает философию Интернета как среды чтения и записи, что хорошо отражено в наблюдении Тима Бернерса-Ли: *«Если вы думаете, что серфинг по гипертексту — это круто, это потому, что вы не пробовали его писать»*.

Чтобы сделать Интернет более доступным (потому что почти ни у кого не было машины NeXT), в CERN был разработан второй проект браузера: **браузер линейного режима «Line Mode»**. Браузер был впервые выпущен в 1991 году и был совместим с большинством систем Unix / Linux.

В 1993 г. программист *Роберт Каулау (Robert Kailau)*, коллега Бернса Ли, сделал **браузер «Samba»** для компьютеров Mac. В том же году сотрудник Института правовой информации (*Legal Information Institute, LI*) *Томас Брюс (Thomas R. Bruce)* распространил первый **браузер «Cello»** для компьютеров класса IBM PC, после чего множество компаний начали выпускать собственные Интернет-навигаторы. Это привело к массовой нестыковке,

потому что каждый производитель браузеров старался дополнить тогда еще крайне ограниченный HTML своим набором тегов.

Ведущие информационные корпорации, недовольные неразберихой тегов HTML, сформировали в декабре 1994 года **Консорциум Всемирной паутины** (*World Wide Web Consortium, W3C*), взявший под свой контроль *работу практически над всеми стандартами важнейших технологий Сети*. Надо отметить, что формально W3C выпускает только рекомендации и некоторые компании их игнорируют, но в целом рекомендации W3C признаются всем IT-рынком в качестве стандартов. В настоящее время главой Консорциума является сэр Тимоти Бернерс-Ли. Сейчас в Консорциум входят более 400 компаний-участников: Adobe, AOL, Apple, Canon, CERN, Cisco, Dow Jones, Google, IBM, Intel, Microsoft, Mozilla, Nokia, Opera, Oracle, Samsung, Siemens, Yahoo и т. д. Все участники Консорциума занимаются разработкой и продвижением открытых стандартов и рекомендаций для нескольких десятков технологий: HTML, XHTML, CSS, HTTP, URI, XML, DOM, MathML, PNG, SVG, XSLT и пр.

На рис. 15 приведена хронология развития языка HTML. **Первая версия языка (HTML 1.0)** была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер.

Вторая версия языка (HTML 2.0) фиксировала практику использования его конструкций. Эта версия предоставляла новые возможности, расширяя набор тегов HTML в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста.

Версия 3.2 смогла упорядочить все нововведения и согласовать их с существующей практикой. HTML 3.2 позволяет реализовать использование таблиц, выполнение кодов языка Java, отображение графики текстом, а также отображение верхних и нижних индексов.

Версия 4.01 включала дополнительные средства работы с мультимедиа, языки программирования, таблицы стилей, упрощенные средства печати изображений и документов. Для управления сценариями просмотра страниц сайта можно было использовать языки программирования сценариев, например, JavaScript.

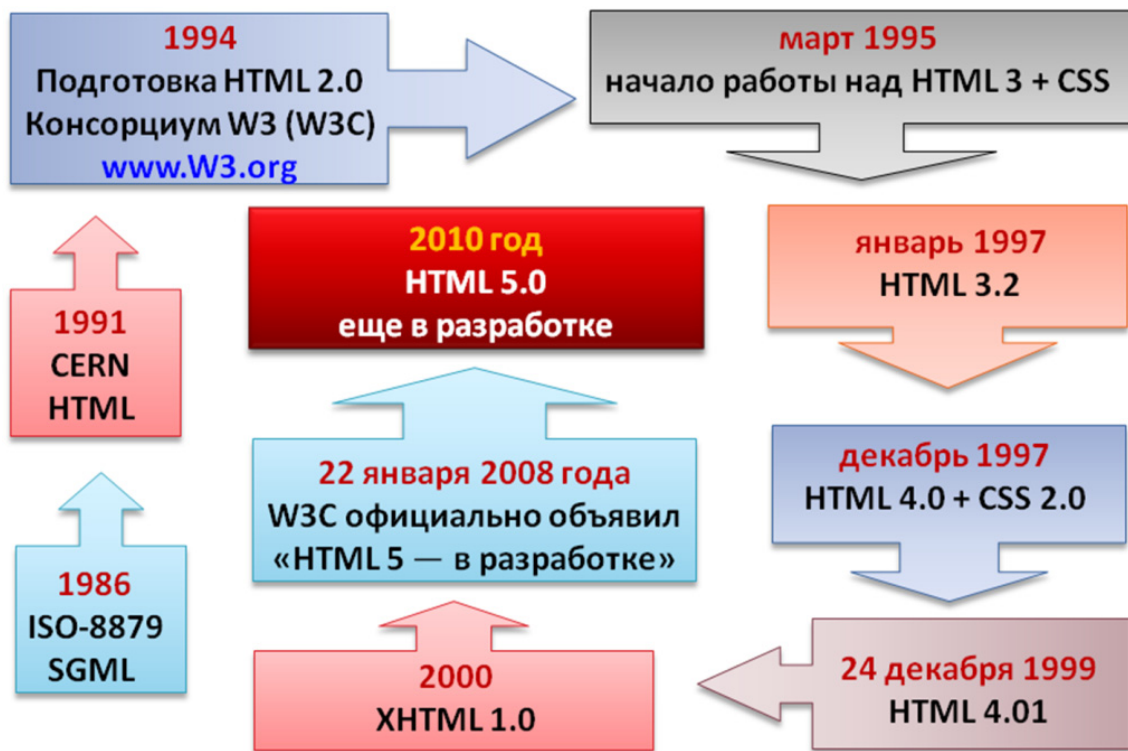


Рис. 15. Хронология развития версий языка HTML

Версия 5 вводит несколько новых элементов и атрибутов, которые отражают типичное использование разметки на современных сайтах. Некоторые устаревшие элементы, которые можно было использовать в HTML 4.01, были исключены, включая чисто оформительские элементы, такие как `` и `<center>`, чьи эффекты выполняются с помощью каскадных таблиц стилей CSS. Синтаксис HTML 5 больше не базируется на SGML, несмотря на подобие его разметки. Также получили развитие языки, разрабатываемые параллельно с HTML: VRML (англ. *Virtual Reality Modelling Language* — язык моделирования виртуальной реальности) и XML (англ. *eXtensible Markup Language* — язык расширенной разметки).

Версия HTML5 до сих пор еще не получила статус официальных рекомендаций W3C, но уже сейчас понятно, что авторы HTML продолжают работать в направлении разработки требований к поддержке объектной модели документа и интерпретации языка JavaScript.

Язык HTML, на котором написано большинство веб-страниц, позволяет передавать размеченное содержание от источника к получателю в очень компактной форме. Такой принцип организации, а именно разделение формы и содержания веб-страниц, лежит в основе связки HTML + CSS.

6. Язык гипертекстовой разметки

Разработчикам языка HTML удалось решить две ключевые задачи:

- предоставить дизайнерам гипертекстовых баз данных простое средство создания документов;
- сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет **выбора теговой модели описания документа**. Такая модель широко применяется в системах подготовки документов для печати.

Язык HTML позволяет размечать электронный документ, который может содержать самые разнообразные метки, иллюстрации, аудио- и видеофрагменты и т. д. В состав языка вошли развитые средства для создания различных уровней заголовков, шрифтовых выделений, различные списки, таблицы, текстовые блоки и многое другое.

Вторым важным моментом, повлиявшим на судьбу HTML, стало то, что **в качестве основы был выбран обычный текстовый файл**.

Таким образом, гипертекстовая база данных в концепции WWW — это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки).

Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео (гипермедиа).

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В World Wide Web функции интерпретатора разделены между веб-сервером гипертекстовой базы данных и интерфейсом пользователя (модель «Клиент-Сервер»). Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, обеспечивает предпроцессорную обработку документов, в то время как интерфейс пользователя осуществ-

ляет интерпретацию конструкций языка, связанных с представлением информации.

Как было отмечено ранее, язык HTML является приложением («частным случаем») SGML (англ. *Standard Generalized Markup Language* — стандартный обобщенный язык разметки) и соответствует международному стандарту ISO 8879.

HTML (от англ. *HyperText Markup Language* — язык гипертекстовой разметки) — стандартный язык разметки документов во Всемирной паутине.

Подавляющее большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме. В HTML используется теговая модель описания документа как совокупности элементов, каждый из которых окружен тегами.

Элемент — конструкция языка HTML, предписывающая способ интерпретации помещенных внутри нее данных.

Тег — единица разметки, стартовый или конечный маркер элемента. Теги определяют границы действия элементов.

Атрибут — параметр или свойство тега. Все атрибуты записываются внутри стартового тега и разделяются пробелами.

По своему значению и принципам использования **теги близки к понятию скобок begin/end** в языках программирования. В HTML они обрамляются угловыми скобками «<» и «>».

Теги определяют область действия правил интерпретации текстовых элементов документа. Элементы документа, размеченного при помощи HTML, также принято называть **контейнерами**. Общая схема построения элемента в формате HTML приведена на рис. 16.

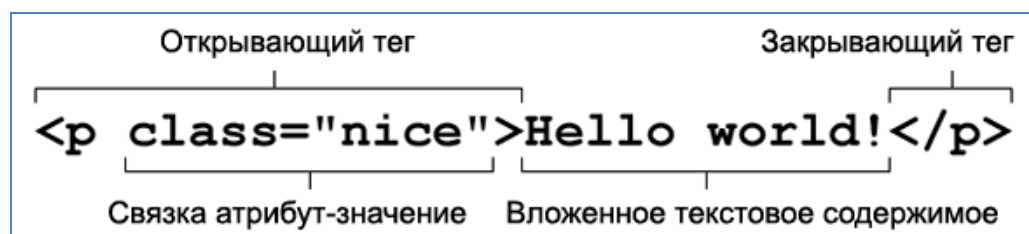


Рис. 16. Элемент языка HTML включает в себя открывающий и закрывающий теги, группу атрибутов со значениями внутри открывающего тега и контент – содержимое (как правило, текстовое) элемента, размещенного между тегами

В общем случае, элемент, маркированный тегом, имеет следующую структуру (имя тега соответствует имени элемента, т. е. в данном случае можно говорить о теге `<element>` либо об элементе с именем `element` — в книге будут использоваться оба эти обращения при описании элементов):

```
<element attr1="value1" attr2="value2"...>  
    внутреннее содержание  
</element>
```

Если тег не имеет внутреннего содержания, то он может иметь упрощенную структуру:

```
1 <element attr1="value1" attr2="value2"...>  
2 <element attr1="value1" attr2="value2"... />
```

В строке 2 в закрывающем теге использована конструкция `</>`, а не просто `<>`. Это необходимо, если используется версия языка XHTML 1.0 — в этом случае элементы должны закрываться (признак закрывающего тега — косая черта (слеш) перед именем тега), даже если использование закрывающего тега запрещено (естественно, ввиду отсутствия содержимого элемента). При этом во избежание ошибок при обработке документа перед этим сочетанием символов нужно вставить пробел.

В языке HTML имеют место элементы, например, `img` и `hr`, в которых запрещен закрывающий тег в виду отсутствия содержимого (контента):

```
  
<hr>
```

Теговая модель описывает документ как совокупность контейнеров, *каждый из которых начинается и заканчивается тегами*, то есть html-документ представляет собой не что иное, как обычный ASCII-файл, с добавленными в него управляющими html-кодами (тегами). Поскольку HTML произошел от SGML, в нем разрешено использовать только *три управляющих символа*: горизонтальную табуляцию, перевод каретки и перевод строки.

Это облегчает взаимодействие html-документа (файла) с различными операционными системами.

Теговая модель также определяет **принцип вложенности элементов** с одним правилом: закрывающие теги вложенных элементов должны встречаться в разметке в обратном порядке относительно одноименных открывающих тегов (пересечение / наложение элементов запрещено). Например,

```
1 <div id="photomag">
2   <a href="ivanenko_big.jpg">
3     
5   </a>
6 </div>
```

В строке 1 начинается элемент `div` (открывающий тег `<div>` с атрибутом `id`). В него вложен элемент `a` (открывающий тег `<a>` с атрибутом `href` в строке 2), в который, в свою очередь вложен элемент `img` (единственный тег `` в строке 3 с атрибутами `src`, `title` и `alt`). Заккрытие элементов происходит строго по правилу теговой модели: так как у элемента `img` нет закрывающего тега, то сначала закрывается элемент `a` (строка 5), а потом элемент `div` (строка 6). Если условно убрать все атрибуты, то приведенная выше разметка будет выглядеть следующим образом:

```
1 <div>
2   <a>
3     <img>
4
5   </a>
6 </div>
```

Здесь более наглядно просматривается принцип вложенности, выраженный в очередности закрывающих тегов по отношению к открывающим.

7. Технология «Клиент-Сервер»

«Клиент-сервер» (англ. *client-server*) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми *серверами*, и заказчиками услуг, называемыми *клиентами*.

Фактически клиент и сервер — это программное обеспечение. Взаимодействие клиента и сервера происходит через компьютерную сеть и производится, как правило, на различных физических устройствах.

Программы-серверы ожидают от программ-клиентов запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством протоколов HTTP, FTP, потоковое мультимедиа или работа с базами данных) или в виде сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями или просмотр веб-страниц). Как правило, одна программа-сервер выполняет запросы от множества программ-клиентов, ее размещают на специально выделенной *вычислительной машине* (VM) с высокой производительностью. Из-за особой роли такой VM, ее также называют сервером, а VM, выполняющие клиентские программы, соответственно, клиентами.

На рис. 17 приведена схема клиент-серверной технологии. Со стороны клиентской части, в качестве инициатора HTTP-запросов к серверу, выступает браузер, обладающий модулями для обработки поступающих html-документов, а также способного выполнить сценарии JavaScript и обработать стили CSS (этимися качествами обладают все без исключения современные браузеры). При поступлении запроса, веб-сервер начинает его обрабатывать путем формирования ответа (в большинстве случаев это html-документы) обращаясь к внутренним ресурсам (серверные модули для «сборки» динамических веб-страниц, базы данных, вспомогательные приложения и т. д.). Таким образом, эта достаточно простая технология постоянно работает в режиме «запрос — ответ».

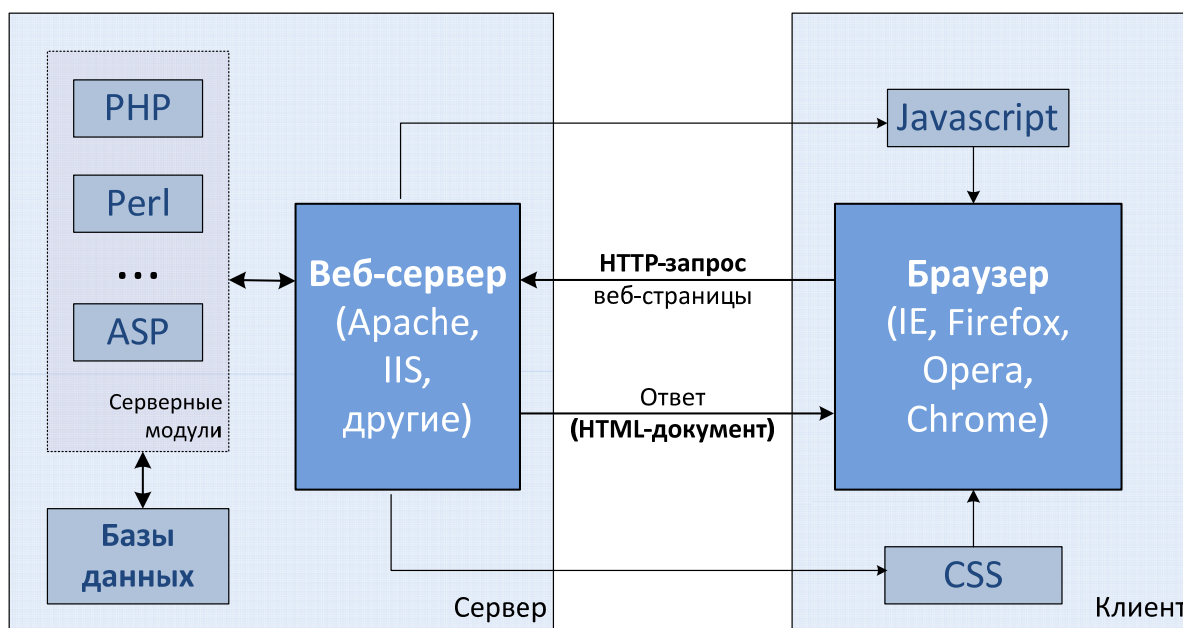


Рис. 17. Клиент-серверная технология

Преимущества технологии «Клиент-сервер»:

- в большинстве случаев возможно распределение вычислительной системы между несколькими независимыми ВМ;
- повышенная защита сервера, на котором хранятся данные, по отношению к защите большинства ВМ клиентов.
- простота организации контроля полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки технологии «Клиент-сервер»:

- неработоспособность сервера (отказ от обслуживания ввиду недостатка производительности, либо физический износ) способна сделать неработоспособной всю вычислительную сеть;
- поддержка работы данной системы требует отдельного специалиста — системного администратора;
- высокая стоимость сетевого оборудования.

8. Обработка веб-документов в браузере

Основное предназначение браузера — отображение веб-документов. Для этого, согласно технологии «Клиент-сервер», от клиента на сервер отправляется запрос, а результат выводится в окне браузера.

Вывод запрошенного содержимого на экране монитора компьютера клиента осуществляется с помощью специального **модуля отображения** в браузере (рис. 18). По умолчанию он способен отображать html- и xml-документы, а также графические файлы. Специальные подключаемые модули (расширения для браузеров) делают возможным отображение другого содержания, например PDF-файлов.

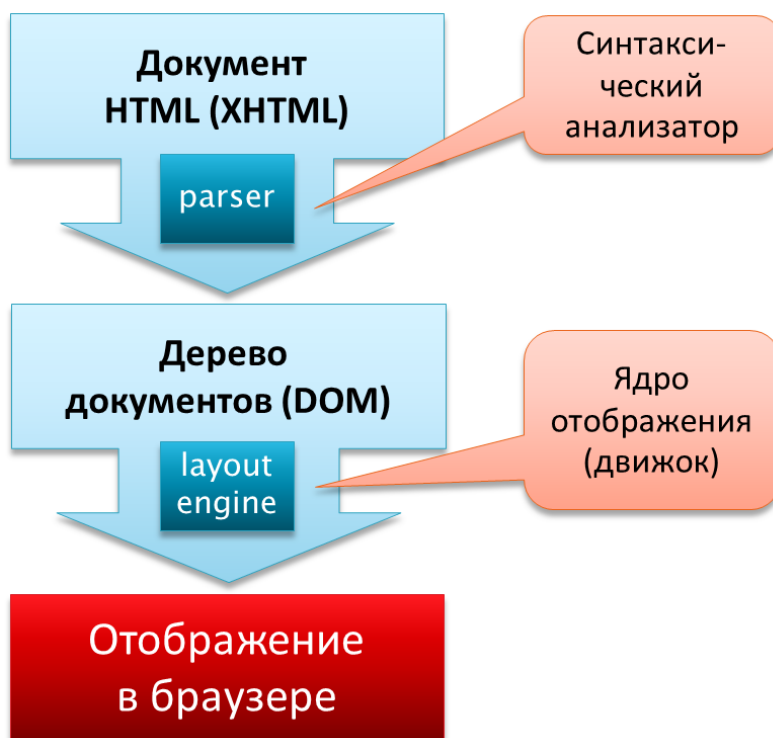


Рис. 18. Модуль отображения веб-документов в браузере

Модуль отображения выполняет **синтаксический анализ веб-документа** и переводит теги в узлы **объектной модели документа DOM** (от англ. *Document Object Model*). Также извлекается информация о стилях как из внешних css-файлов, так и из одноименных элементов и атрибутов `style` (см. п. 2.2.3). Загруженные и распознанные стили формируют **модель CSSOM** (англ. *CSS Object Model*). Эта информация и инструкции по отображе-

нию в html-файле используются для создания еще одного дерева — **дерева отображения** (дерево рендеринга, или *render tree*). Оно содержит узлы-блоки с визуальными атрибутами, такими как цвет и размер. Узлы-блоки располагаются в том порядке, в каком они должны быть выведены на экран.

После создания дерева отображения начинается **компоновка элементов** (*layout*), в ходе которой каждому узлу присваиваются координаты точки на экране, где он должен появиться. Затем выполняется отрисовка, при которой узлы дерева отображения **последовательно отрисовываются** (браузеры используют поточный метод (*flow*), при котором в большинстве случаев достаточно одного прохода для размещения всех элементов) с помощью исполнительной части пользовательского интерфейса.

Важно понимать, что это последовательный процесс. Для удобства пользователя модуль отображения старается вывести содержание на экран как можно скорее, поэтому создание дерева отображения и компоновка могут начаться еще до завершения синтаксического анализа кода HTML. Одни части документа анализируются и выводятся на экран, в то время как другие только передаются по сети.

В процессе взаимодействия пользователя со страницей, а также выполнения скриптов, последовательность отображения веб-документов в браузере меняется, что требует повторного выполнения некоторых из вышеперечисленных операций. К таким операциям относятся переотрисовка и перерасчет положения на странице.

Переотрисовка (*repaint* или *restyle*) происходит в случае изменения стилей произвольного элемента, не влияющих на его размеры и положение на странице (например, `background-color`, `border-color`, `visibility`). При этом браузер просто отрисовывает его заново, с учетом нового стиля.

Перерасчет положения на странице (*reflow* или *relayout*) происходит, если изменения затрагивают содержимое, структуру документа, положение элементов. Причинами таких изменений обычно являются:

- манипуляции с DOM (добавление, удаление, изменение, перестановка элементов);
- изменение содержимого текста в полях форм;
- расчет или изменение css-свойств;

- добавление, удаление таблиц стилей;
- манипуляции с атрибутом `class` (отработка скриптов);
- манипуляции с окном браузера (изменение размеров, прокрутка страницы);
- активация псевдо-классов (например, `:hover`).

Браузеры по возможности локализуют *repaint* и *reflow* в пределах элементов, подвергнувшимся изменению. Например, изменение размеров абсолютно или фиксировано спозиционированного элемента затронет только сам элемент и его потомков, в то время как изменение статично спозиционированного — повлечет *reflow* всех элементов, следующих за ним.

Еще одна особенность — во время выполнения JavaScript браузеры сохраняют в кэше² вносимые изменения, и применяют их в один проход по завершению работы блока кода.

Детально разберемся с упомянутыми выше понятиями. Прежде всего, это модули отображения (ядро или движок браузера), которые, по сути, позиционируются как авторские программные продукты (ноу-хау) браузерных компаний и в большинстве случаев являются коммерческой тайной.

В Firefox применяется модуль **Gecko** — собственная разработка компании Mozilla, а в браузерах Safari и Chrome используется движок **WebKit**. WebKit, совместно разработанный компаниями Apple, Adobe, KDE и др., представляет собой модуль отображения с открытым исходным кодом, который был изначально разработан для платформы Linux и адаптирован компанией Apple для Mac OS и Windows. Кроме Gecko и WebKit, в настоящее время активно разрабатываются следующие движки (в скобках указана компания-разработчик): **Blink** (Google), **Goanna** (Mozilla), **KHTML** (KDE), **Prince** (YesLogic), **Servo** (Mozilla), **EdgeHTML** (Microsoft).

Следующее понятие, являющееся важным этапом работы модуля отображения — синтаксический анализ (англ. *parsing* —

² Кэш-память (или просто — кэш) браузера содержит все документы, загруженные пользователем по протоколу HTTP. Используется для доступа к ранее загруженным страницам при навигации назад/вперед, позволяет сохранять страницы, или просматривать их код, не обращаясь повторно к серверу. Настройки кэш-памяти (например, очистка кэша) доступны пользователю в настройках браузера.

парсинг), — преобразование веб-документа в пригодную для чтения и выполнения структуру. Результатом синтаксического анализа, как правило, является дерево узлов, представляющих структуру документа. Оно называется деревом синтаксического анализа, или просто синтаксическим деревом.

Синтаксический анализ работает на основе определенных правил, которые определяются языком (форматом) документа. Для каждого формата существуют грамматические правила, состоящие из словаря и синтаксиса. Они образуют т. н. бесконтекстную грамматику, в которой грамматические правила могут быть полностью выражены в формате BNF (форма Бэкуса — Наура, англ. *Backus Normal Form*).

Задача синтаксического анализатора HTML — переводить информацию из кода HTML в синтаксическое дерево. Словарь и синтаксис языка HTML определены в спецификациях W3C.

Однако HTML (в отличие от XHTML, CSS и JavaScript) невозможно определить с помощью бесконтекстной грамматики, с которой работают синтаксические анализаторы, поскольку грамматика формального стандарта определения HTML — формата DTD (*Document Type Definition*), — не является бесконтекстной. Это связано с «мягким» синтаксисом HTML — способность языка «прощать» ошибки (вплоть до пропуска открывающих или закрывающих тегов, которые подставляются автоматически) ощутимо облегчает жизнь разработчику. С другой стороны, из-за этого становится сложно формально определить грамматику.

Поэтому определение HTML включено в формат DTD (содержит определения всех допустимых элементов, их атрибутов и иерархии), в котором не задается бесконтекстная грамматика. При этом существует несколько версий DTD.

Полученное синтаксическое дерево состоит из элементов DOM и узлов атрибутов. **DOM** (от англ. *Document Object Model* — «объектная модель документа») — специальная, языково-независимая интерфейсная модель разбора документов XML и HTML. DOM также служит для представления html-документа и интерфейса элементов HTML таким внешним объектам, как код JavaScript.

Модель DOM практически идентична разметке. В самой разметке следует различать **символьные данные** (*character data*) и **компоненты разметки** (*markup*). Символьные данные пред-

ставляют информацию, которая будет отображена в окне браузера, в то время как компоненты разметки (невидимы для конечного пользователя через окно браузера) определяют структуру этого отображения.

Рассмотрим пример разметки. Здесь темно-серым цветом выделены только символьные данные:

```
<p><a href="http://www.w3.org/">Консорциум W3C</a>  
Разрабатывает<strong>стандарты</strong> HTML и  
CSS.</p>
```

Эта же разметка с выделенными компонентами разметки (открывающие и закрывающие теги):

```
<p><a href="http://www.w3.org/">Консорциум W3C</a>  
Разрабатывает<strong>стандарты</strong> HTML и  
CSS.</p>
```

Объектная модель приведенного выше фрагмента кода HTML представлена на рис. 19. В ней, согласно разметке, имеют место узлы элемента и текстовые узлы. Между узлами элементов существуют определенные отношения.

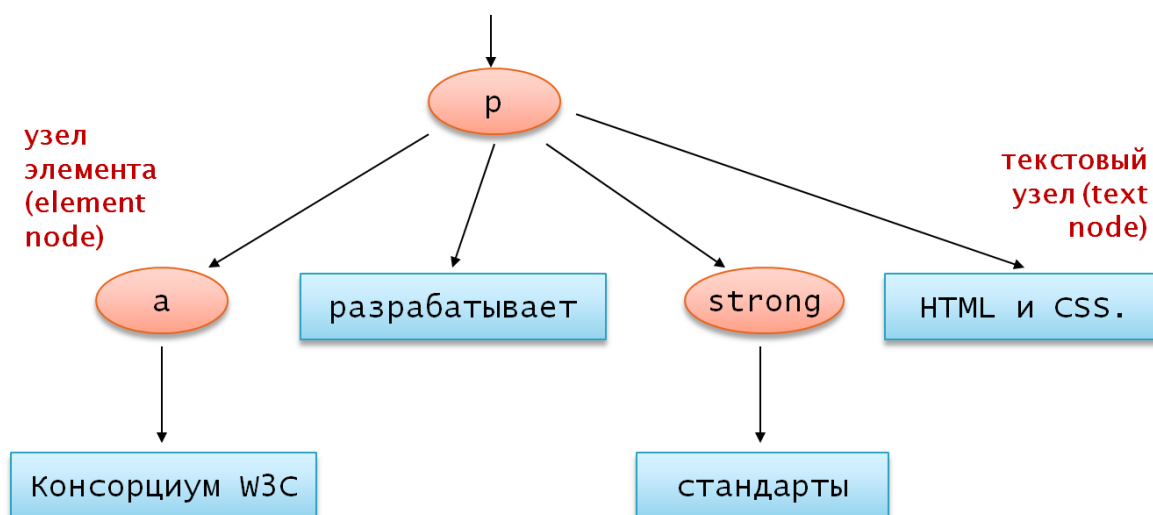


Рис. 19. Объектная модель фрагмента кода

9. Структура документа HTML

Любой гипертекстовый документ HTML состоит из следующих обязательных частей (рис. 20):

1. Версия html-документа (**DOCTYPE**).
2. Обозначение языка документа (**HTML**).
3. «Голова» документа (**HEAD**).
4. Заголовок окна веб-страницы (**TITLE**).
5. «Тело» документа (**BODY**).

На рис. 21 приведен пример разметки с обязательными элементами и построенной объектной моделью документа.

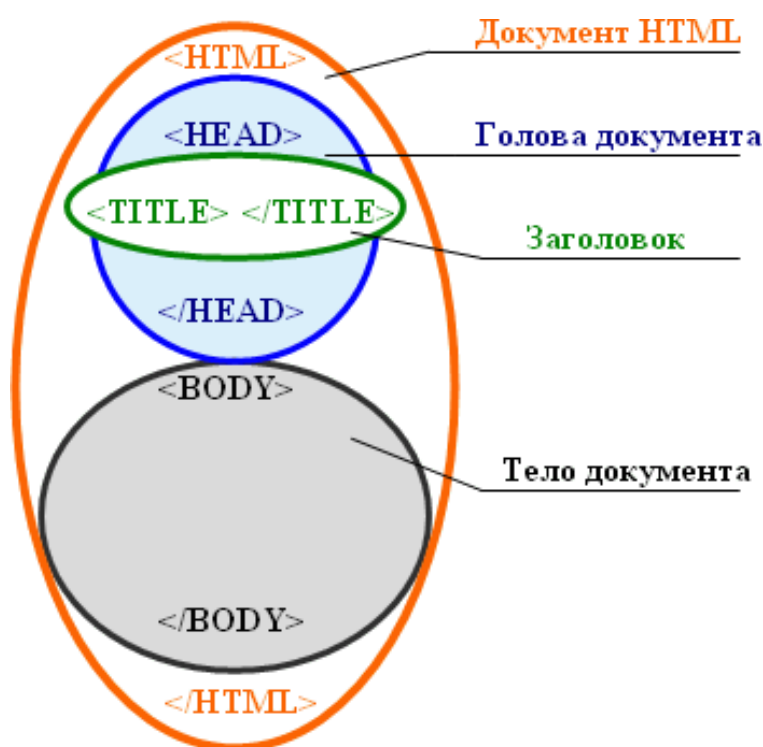


Рис. 20. Структура документа HTML

Детально рассмотрим обязательный элемент html-разметки, определяющий тип текущего документа.

Элемент `<!DOCTYPE>` предназначен для указания типа текущего документа — DTD. Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу. Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу, необходимо в первой строке кода прописывать `<!DOCTYPE>`. При этом `<!DOCTYPE>` не участвует в построении DOM (см. рис. 21).

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>My first page</title>
  </head>
  <body>
    <p>Hello, World!</p>
  </body>
</html>

```

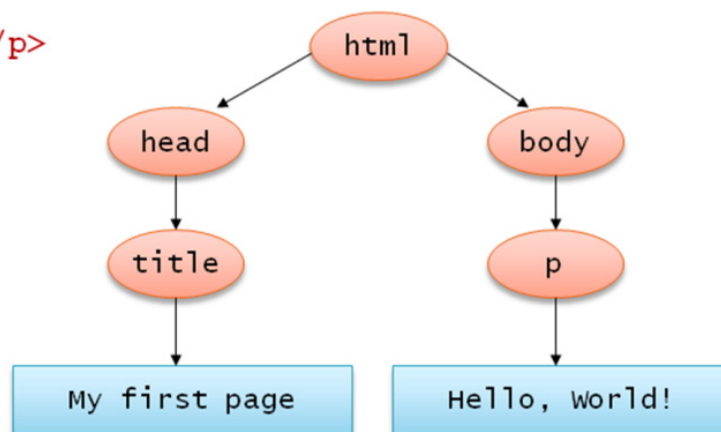


Рис. 21. Пример разметки HTML с обязательной структурой и объектной моделью документа

Существует несколько видов `<!DOCTYPE>`, они различаются в зависимости от версии языка, на который ориентированы. В таблице 1.1 приведены основные типы документов с их описанием.

Синтаксис элемента `<!DOCTYPE>`:

```

<!DOCTYPE [Элемент верхнего уровня]
[Публичность]
"[Регистрация]//[Организация]//[Тип]
[Имя]//[Язык]"
"[URL]">

```

Синтаксис включает следующие параметры:

✓ *Элемент верхнего уровня* — указывает элемент верхнего уровня в документе, для HTML это тег `<html>`.

✓ *Публичность* — объект является публичным (значение `PUBLIC`) или системным ресурсом (значение `SYSTEM`), например, таким как локальный файл. Для HTML/XHTML указывается значение `PUBLIC`.

✓ *Регистрация* — сообщает, что разработчик DTD зарегистрирован в международной организации по стандартизации (*International Organization for Standardization, ISO*). Принимает одно из двух значений: плюс «+» — разработчик зарегистрирован

в ISO и минус «-» — разработчик не зарегистрирован. Для W3C значение ставится «-».

✓ *Организация* — уникальное название организации, разработавшей DTD. Официально HTML/XHTML публикует W3C, это название и пишется в `<!DOCTYPE>`.

Таблица 1. – Допустимые форматы DTD

DOCTYPE	Описание
HTML 4.01	
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>	строгий синтаксис HTML
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	переходный синтаксис HTML
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></code>	в документе HTML применяются фреймы
HTML 5	
<code><!DOCTYPE html></code>	для всех документов
XHTML 1.1	
<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"></code>	строгий синтаксис XHTML

✓ *Тип* — тип описываемого документа. Для HTML/XHTML указывается значение DTD.

✓ *Имя* — уникальное имя документа для описания DTD.

✓ *Язык* — язык, на котором написан текст для описания объекта. Содержит две буквы, пишется в верхнем регистре. Для документа HTML/XHTML указывается английский язык (EN).

✓ *URL* — адрес документа с DTD.

При **строгом синтаксисе** HTML, описанном в файле **strict.dtd**, допускается включать все теги и атрибуты, кроме *осуждаемых*. Правила синтаксиса зависят от используемой версии, но в целом должны соблюдаться следующие:

1. Осуждается использование следующих тегов: `<applet>`, `<basefont>`, `<center>`, `<dir>`, ``, `<isindex>`, `<noframes>`,

`<plaintext>`, `<s>`, `<strike>`, `<u>`, `<xmp>`. Взамен по возможности рекомендуется использовать стили.

2. Текст, изображения и элементы форм нельзя напрямую добавлять в `<body>`, эти элементы должны обязательно находиться внутри блочных элементов `<p>` или `<div>`.

3. Осуждается применение атрибутов `target`, `start` (тег ``), `type` (теги ``, ``, ``) и др.

Естественно, также должен строго соблюдаться синтаксис языка — правильное вложение тегов, закрытие тегов, должны присутствовать обязательные теги и др.

При **переходном синтаксисе** HTML, описанном в файле **loose.dtd**, применяется «мягкий» синтаксис языка, в котором допускается использовать все теги и атрибуты, включая осуждаемые. Цель переходного `<!DOCTYPE>` заключается в постепенном знакомстве с синтаксисом языка. Если сразу использовать строгий синтаксис начинающему веб-разработчику покажется слишком сложным, для него и предназначен переходный синтаксис HTML.

Синтаксис с фреймами в HTML, описанном в файле **frameset.dtd**, применяется во фреймовой структуре в документе, и аналогичен переходному синтаксису. Данный `<!DOCTYPE>` применяется только для главного документа, формирующего структуру фреймов с помощью тегов `<frameset>` и `<frame>`.

В HTML5 существует лишь один тип `<!DOCTYPE html>`, который переводит браузер в стандартный режим (строгий синтаксис HTML).

10. Каскадные таблицы стилей. История развития CSS

Каскадные (многоуровневые) таблицы стилей CSS являются стандартом на основе текстового формата, определяющего представление данных в браузере.

CSS (англ. *Cascading Style Sheets* — *каскадные таблицы стилей*) — разработанный Консорциумом Всемирной паутины (W3C) формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Прежде всего, CSS используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языка разметки HTML, но может также применяться к любым XML³-документам, например, к SVG⁴.

Основная цель разработки CSS — разделение описания логической структуры веб-страницы от описания ее внешнего вида и гибкого управления отображением гипертекстовых документов, что:

- приводит к увеличению доступности документа;
- предоставляет большую гибкость и возможность управления представлением документа;
- уменьшает сложность и повторяемость в структурном содержимом.

Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть.

Перед изучением CSS необходимо определиться с некоторыми понятиями, связанными с каскадными таблицами:

1) *Стиль* — совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стиль

³ XML (*eXtensible Markup Language*) — расширяемый язык разметки от консорциума W3C, являющийся удобным (с точки зрения простоты синтаксиса) для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком [1].

⁴ SVG (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики, от консорциума W3C, который входит в подмножество расширяемого языка разметки XML и предназначен для описания двумерной векторной и смешанной векторно-растровой графики в формате XML.

включает практически все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

2) *Таблица стилей* — совокупность стилей, применимых к гипертекстовому документу.

3) *Каскадирование* — порядок применения различных стилей к веб-странице. Здесь требуются некоторые разъяснения. Браузер последовательно применяет стили в соответствии с приоритетом их подключения. Другой аспект каскадирования — *наследование (inheritance)*, — конкретный стиль применим ко всем дочерним элементами гипертекстового документа.

Официальная информация о **спецификации Cascading Style Sheets** доступна на сайте Консорциума W3C.

Удобство и универсальность каскадных таблиц стилей также обусловлено его гибкостью: CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как:

- экранное представление;
- «мобильное» представление;
- печатное представление;
- чтение голосом (специальным голосовым браузером);
- при выводе устройствами, использующими шрифт Брайля.

CSS — это широкий спектр технологий, который одобрен консорциумом W3C и имел название «Веб Стандарты». В 90-х годах прошлого века назрела необходимость стандартизировать их в какие-то определенные единые правила, благодаря которым веб-дизайнеры и программисты проектировали бы сайты.

Вначале различные браузеры имели свои собственные стили для отображения веб страниц. HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, поэтому CSS не получил тогда широкого признания.

Термин «*каскадные таблицы стилей*» предложил **Хокон Виум Ли**⁵ в 1994 году. Совместно с **Бертом Босом**⁶ он стал раз-

⁵ Хокон Виум Ли (норв. *Håkon Wium Lie*) (род. в 1965 г.) — норвежский ученый, специалист в области информатики. В 1994 г. работал на W3C, INRIA, CERN. С 1999 года работает в норвежской компании Opera Software, известной благодаря браузеру Opera. В 2006 году в Университете Осло защитил диссертацию по теме «Каскадные таблицы стилей».

вивать CSS. В 1996 году в консорциуме W3C создается группа разработчиков CSS (CSS Working Group), председателем которой становится Берт Бос.

Первая версия **CSS1 (уровень 1)** в качестве рекомендации от W3C принята 17 декабря 1996 года. Ее основные возможности:

- управление способом отображения элемента на странице;
- возможность для элемента задать и запретить обтекание текстом;
- управление размерами, внешними и внутренними отступами элемента;
- управление вертикальным выравниванием в табличных блоках;
- управление границами элемента (стиль, цвет, ширина);
- управление форматированием нумерованных и ненумерованных списков (тип маркера, обтекание маркера текстом, в качестве маркера — изображение);
- возможность задавать цвет текста и цвет фона элемента;
- возможность задавать в качестве фона элемента изображения, а также позиционирование и повторение этого изображения в фоне;
- управление параметрами шрифта (название, размер, курсив и жирность);
- управление свойствами текста (выравнивание, регистр, отступы, оформление);
- управление междустрочным интервалом, и расстоянием между словами и буквами.

Через полтора года, 12 мая 1998 года, принята вторая версия **CSS2 (уровень 2)** все так же в качестве рекомендации от W3C. В дополнение к CSS1, вторая версия обладала обратной совместимостью и новыми возможностями, позволяющими:

⁶ Гийсберт (Берт) Бос (англ. *Gijsbert (Bert) Bos*) (род. в 1963 г.) — ученый, специалист в области компьютерных наук. Изучал математику в университете Гронингена (University of Groningen), Нидерланды. В 1996 году присоединился к консорциуму W3C для работы над каскадными таблицами стилей. Бывший председатель и нынешнее контактное лицо W3C в рабочей группе CSS.

- управлять направлением текста в элементе (слева направо или справа налево);
- задавать видимую область элемента и обрезать все остальное;
- управлять отображением контента за пределами размеров элемента;
- управлять внешним видом курсора;
- управлять положением элементов по оси z (один элемент поверх другого);
- показывать вместо элемента пустое место;
- задавать минимально возможные и максимально возможные размеры элемента;
- указывать расстояние между ячейками таблицы, либо схлопывать их;
- управлять обводкой элемента: задавать ее толщину, тип и цвет;
- указать тип и цвет для каждой границы элемента отдельно;
- задавать фиксированные размеры элементам таблицы;
- управлять внешним видом кавычек;
- задавать таблицы стилей для не визуальных носителей: управлять контентом при печати, а также задавать звуковое оформление контента (силу, громкость голоса, длину пауз и т. д.) для голосовых браузеров.

Для следующих изменений, коснувшихся CSS2, потребовалось более 10-ти лет — 8 сентября 2009 года была принята версия **CSS 2.1 (уровень 2, ревизия 1)**. По сути CSS 2.1, это не что иное, как результат приведения спецификации CSS2 в соответствие со сложившимися к тому времени реалиями. В результате в CSS2.1:

- исправлен ряд ошибок CSS2;
- изменены некоторые моменты, реализация которых в подавляющем большинстве браузеров отличается от спецификации CSS2;
- убраны особенности CSS2, которые, в силу того, что не были реализованы, были отвергнуты CSS-сообществом;
- удалены фрагменты CSS2, которые будут устаревшими в CSS3;

– добавлены некоторые новые значения свойств.

CSS3 (уровень 3) — активно разрабатываемая спецификация CSS. Это самая масштабная редакция по сравнению с CSS1, CSS2 и CSS2.1. Главной особенностью CSS3 является возможность создавать анимированные элементы без использования языка сценариев JavaScript. В числе прочего: поддержка линейных и радиальных градиентов, теней, различных скруглений, множественных фонов и пр.

CSS4 (уровень 4) разрабатывается консорциумом W3C с 29 сентября 2011 года. Модули CSS4 построены на основе CSS3 и дополняют их новыми свойствами и значениями. Все они существуют пока в виде черновиков (working draft).

Как и с последними версиями HTML, версии CSS3 и CSS4 неоднозначны по отношению друг к другу. Так как CSS3 — долгое время разрабатываемая версия, то ее как бы «перекрывает» уже следующая — CSS4.

11. Контрольные вопросы

1. Дайте определения понятиям «гипертекст» и «гипертекстовая система».
2. Дайте определения понятиям «гиперсвязь» и «гиперссылка».
3. Что включает в себя понятие «гипермедиа» по отношению к гипертексту?
4. В чем заключается основная идея гипертекстовых систем?
5. Перечислите основные достижения в развитии технологии гипертекста в разрезе трех поколений его исторического развития.
6. Перечислите ранние версии языка HTML до 4.01. Какие дополнения к языку разметки привносила каждая версия?
7. Опишите принципы теговой модели. Дайте определения понятиям «элемент», «тег» и «атрибут».
8. В чем заключается принцип вложенности элементов?
9. Приведите определение клиент-серверной технологии.
10. Какими преимуществами и недостатками обладает технология «Клиент-сервер»?
11. Из каких этапов состоит процесс обработки веб-документов в браузере?
12. Дайте определение объектной модели документа. Как выделить такие символичные данные и компоненты разметки в коде HTML?
13. Из каких структурных элементов (обязательных частей) состоит документ HTML?
14. Перечислите свойства строгого и переходного синтаксиса HTML. В каких случаях необходимо применять строгий синтаксис, а в каких — переходный?
15. Как строится дерево документов? Перечислите родственные связи между узлами элементов.?
16. Дайте определение каскадным (многоуровневым) таблицам CSS. Как расшифровывается аббревиатура «CSS»?
17. В чем заключается основная цель разработки CSS?

Рекомендуемая литература:

Основная:

1. Методические указания к лабораторным занятиям по дисциплине «Интернет-технологии» [Электронный ресурс] – для студентов уровня профессионального образования «магистр» всех направлений подготовки и форм обучения / ГОУВПО «ДОННТУ», каф. компьютерной инженерии; сост. А. Я. Аноприенко, С. В. Иваница, К. А. Сидоров – Электрон. дан. (1 файл: 1,99 Мб). – Донецк: ДОННТУ, 2020. – Систем. требования: Acrobat Reader.
2. Методические указания по выполнению индивидуального задания по дисциплине «Интернет-технологии» : для студентов уровня профессионального образования «магистр» всех направлений подготовки и форм обучения / ГОУВПО «ДОННТУ», Каф. компьютерной инженерии ; сост.: А. Я. Аноприенко, С. В. Иваница, К. А. Сидоров. – Донецк: ДОННТУ, 2020. – Систем. требования: Acrobat Reader. – Загл. с титул. экрана.
3. Аноприенко, А. Я. Интернет-технологии для студентов и преподавателей : учеб. пособие. Кн. 1 / А. Я. Аноприенко, С. В. Иваница, Т. В. Завадская. — Донецк : Технопарк ДонНТУ «УНИТЕХ», 2015. – 260 с. : ил.
4. Иваница, С. В. Веб-типографика. Искусство оформления текстов для Интернета. – Донецк: ДонНТУ, УНИТЕХ, 2013. – 384 с.
5. Аноприенко, А. Я. Интернет-технологии для студентов и преподавателей : учеб. пособие для обучающихся образоват. учреждений высш. проф. образования : книга вторая / А. Я. Аноприенко, С. В. Иваница ; ГОУВПО «ДОННТУ». —Донецк : УНИТЕХ, 2021. — 268 с. : ил.
6. Сычев, А. В. Web-технологии : учебное пособие / А. В. Сычев. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 407 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89412.html>.

Дополнительная:

1. Аноприенко, А.Я. Университет в современном информационном пространстве: тенденции, рейтинги и опыт развития портала магистров ДонНТУ // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2011). Выпуск 13 (185). – Донецк: ДонНТУ, 2011. С. 224–235.
2. Семенов, А. А. Сетевые технологии и Интернет : учебное пособие / А. А. Семенов. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 148 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/66840.htm>.
3. Кудряшев, А. В. Введение в современные веб-технологии : учебное пособие / А. В. Кудряшев, П. А. Светашков. — 3-е изд. — Москва, Саратов : Интер-

- нет Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 359 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89430.html>.
4. Граничин, О. Н. Информационные технологии в управлении : учебное пособие / О. Н. Граничин, В. И. Кияев. — 3-е изд. — Москва, Саратов : Интернет-Университет Информ. Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 400 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89437.html>.
 5. Левин, В. И. История информационных технологий : учебник / В. И. Левин. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 750 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89440.html>.
 6. Заика, А. А. Локальные сети и интернет : учебное пособие / А. А. Заика. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 323 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89442.html>.
 7. Кузнецова, Л. В. Современные веб-технологии : учебное пособие / Л. В. Кузнецова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 187 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89473.htm>.
 8. Семенов, Ю. А. Протоколы и алгоритмы маршрутизации в Интернет : учебное пособие / Ю. А. Семенов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 998 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/94862.html>.
 9. Семенов, Ю. А. Процедуры, диагностики и безопасность в Интернет : учебное пособие / Ю. А. Семенов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 581 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/94863.html>.
 10. Барский, А. Б. Параллельные информационные технологии : учебное пособие / А. Б. Барский. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 502 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97573.html>.
 11. Кудряшев, А. В. Введение в современные веб-технологии : учебное пособие / А. В. Кудряшев, П. А. Светашков. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 359 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89430.html>.
 12. Кузнецова, Л. В. Современные веб-технологии : учебное пособие / Л. В. Кузнецова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 187

- с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89473.htm>.
13. Савельев, А. О. Проектирование и разработка веб-приложений на основе технологий Microsoft : учебное пособие / А. О. Савельев, А. А. Алексеев. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 418 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/94860.html>.
 14. Вагин, Д. В. Современные технологии разработки веб-приложений : учебное пособие / Д. В. Вагин, Р. В. Петров. — Новосибирск : Новосибирский государственный технический университет, 2019. — 52 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/98738.html>.
 15. Лучанинов, Д. В. Основы разработки web-сайтов образовательного назначения : учебное пособие / Д. В. Лучанинов. — Саратов : Ай Пи Эр Медиа, 2018. — 105 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/70775.html>.
 16. Титов, В. А. Разработка WEB-сайта средствами языка HTML : учебное пособие / В. А. Титов, Г. И. Пещеров. — Москва : Институт мировых цивилизаций, 2018. — 184 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/80643.html>.
 17. Фролов, А. Б. Web-сайт. Разработка, создание, сопровождение : учебное пособие / А. Б. Фролов, И. А. Нагаева, И. А. Кузнецов ; под редакцией И. А. Нагаевой. — Саратов : Вузовское образование, 2020. — 355 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/93989.html>.
 18. Титов, В. А. Разработка WEB-сайта средствами языка HTML : учебное пособие / В. А. Титов, Г. И. Пещеров. — Москва : Институт мировых цивилизаций, 2018. — 184 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/80643.html>.
 19. Беликова, С. А. Основы HTML и CSS: проектирование и дизайн вебсайтов : учебное пособие по курсу «Web-разработка» / С. А. Беликова, А. Н. Беликов. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2020. — 174 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/100186.html>.
 20. Основы работы с HTML : учебное пособие / . — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 208 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102036.html>.

Учебное издание

Методические указания
к самостоятельной работе
по дисциплине «Интернет-технологии»
для студентов уровня
профессионального образования «магистр»
всех направлений подготовки
и форм обучения ГОУВПО «ДОННТУ»

Составители:

Аноприенко Александр Яковлевич

Иваница Сергей Васильевич

Сидоров Константин Андреевич