

## 1.1. История возникновения и развития Веб

### 1.1.1. Понятие "Интернет"

В наше время нельзя представить человека, не использующего информационные технологии. Но если в двадцатом веке передовыми технологиями считались радио, телевидение и телефонная связь, то в двадцать первом веке на первый план выходит *Интернет*. Гигантские объемы текстовой, аудио-видео информации, новостные ленты, блоги, социальные сети, игры и многое, многое другое. В настоящее время подключиться к *Интернету* можно через спутники связи, радиоканалы, кабельное телевидение, телефон, сотовую связь, специальные оптоволоконные линии или электропровода. Всемирная сеть стала неотъемлемой частью жизни в развитых и развивающихся странах.

В течение всего пяти лет *Интернет* достиг аудитории свыше 50-ти миллионов пользователей. Другим средствам массовой информации потребовалось гораздо больше времени для достижения такой популярности: Радио – 38 лет, Телевидение – 13 лет.

Попробуем разобраться в этом многогранном понятии " *Интернет* ". Рассмотрим определения в различных источниках:

- *Интернет* (Internet, сокр. от *Interconnected Networks* – объединенные сети) [2] – глобальная телекоммуникационная сеть информационных и вычислительных ресурсов. Служит физической основой для Всемирной паутины. Часто упоминается как Всемирная сеть, Глобальная сеть, либо просто Сеть.
- *Интернет* (Internet от international net – международная сеть) [3] – всемирная компьютерная сеть (информационная система связи, объединяющая множество компьютеров во всем мире).
- *Интернет* (Сеть *Интернет*, Internet) – глобальная информационная сеть, части которой логически взаимосвязаны друг с другом посредством единого адресного пространства, основанного на протоколе *TCP/IP* [4]. *Интернет* состоит из множества взаимосвязанных компьютерных сетей и обеспечивает удаленный доступ к компьютерам, электронной почте, доскам объявлений, базам данных и дискуссионным группам.

- *Интернет* (Internet – inter + net – объединение сетей) [5] – всемирная компьютерная сеть, объединяющая миллионы компьютеров в единую информационную систему.

Когда слово *internet* написано со строчной буквы, оно означает просто объединение сетей (*interconnected networks*) посредством маршрутизации пакетов данных [6]. В этом случае не имеется в виду глобальное информационное пространство *Интернет* (Internet). В неанглоязычной или нетехнической среде эти понятия обычно не различают.

Словарь русского языка Российской академии наук под редакцией В. В. Лопатина рекомендует [7] написание слова с прописной буквы: *Интернет*. Написание со строчной буквы используется в сложных словах, таких как "интернет-портал" и "интернет-магазин".

Некоторые издания (Яндекс, РОЦИТ, "Коммерсантъ", "Наука и жизнь", "Студия Артемия Лебедева" [8] и др.) считают, что собственное имя Всемирной сети уже стало нарицательным и пишут "интернет" с маленькой буквы.

Слово " *Интернет* " склоняется по правилам русской грамматики как существительное мужского рода, ничем не отличаясь от таких слов, как интернат и интерфейс. Поэтому писать следует: "в *Интернете* ", "структура *Интернета* " [8, 9]. Однако в настоящее время также распространено письменное употребление слова как несклоняемого: "подключиться к *Интернет* ", "из *Интернет* ".

Далее в лекции слово " *Интернет* " будет употребляться с большой буквы.

В настоящее время, когда слово *Интернет* употребляется в обиходе, чаще всего имеется в виду **Всемирная паутина** ( *World Wide Web* – концепция, предложенная Т.Бернерс-Ли в 1990 [10]) и доступная в ней информация, а не сама физическая сеть.

По данным известной консалтинговой компании IDC [11] к середине 2008 года число пользователей, регулярно использующих *Интернет*, составило около 1,5 млрд. человек (около четверти населения Земли), а объем данных, хранящихся в *Интернете*, вплотную приблизился к отметке в 500 экзабайтов (500 млрд. Гб) весной 2009 г. При этом по прогнозам аналитиков, к 2011 году количество данных вырастет еще в 2 раза, а к 2016 году *Интернет* будет насчитывать более 2 миллиардов пользователей.

Некоторые юридические аспекты *Интернета* [2]:

- У *Интернета* нет собственника, так как он является совокупностью сетей, которые имеют различную географическую принадлежность.
- *Интернет* нельзя выключить целиком, поскольку маршрутизаторы сетей не имеют единого внешнего управления.
- *Интернет* стал достоянием всего человечества.
- В *Интернете* имеется много полезных и вредных свойств, эксплуатируемых заинтересованными лицами.
- *Интернет*, прежде всего, средство открытого хранения и распространения информации. По маршруту транспортировки незашифрованная информация может быть перехвачена и прочитана.
- *Интернет* может связать каждый компьютер с любым другим, подключенным к Сети, так же, как и телефонная сеть.
- Сайты в *Интернете* распространяют информацию по такому же принципу, то есть индивидуально, по инициативе читателя.
- Спам-серверы и "зомби-сети" распространяют информацию по инициативе отправителя и забивают почтовые ящики пользователей электронной почты спамом точно так же, как забивают реальные почтовые ящики распространители рекламных листовок и брошюр.

### 1.1.2. Хронология

Далее в хронологическом порядке опишем некоторые вехи в развитии *Интернета* [12,13, 15, 16]:

**1957 год.** Запуск в СССР первого искусственного спутника Земли – начало технологической гонки между СССР и США, приведшей, в итоге, к созданию глобальной сети *Интернет*.

**1958 год.** В США при Министерстве обороны создано Агентство Передовых Исследовательских Проектов – Advanced Research Projects Agency (ARPA).

**1961 год.** Студент Массачусетского Технологического Института Леонард Клейнрок описывает технологию, способную разбивать файлы на куски и передавать их различными путями через сеть.

**1963 год.** Руководитель компьютерной лаборатории *ARPA* Джон Ликлидер предлагает первую, детально разработанную концепцию компьютерной сети "Galactic Network".

**1967 год.** Ларри Робертс предлагает связать между собой компьютеры *ARPA*. Компьютерная сеть была названа ARPANET (Advanced Research Projects Agency Network).

**29.10.1969 года.** В 21:00 между двумя первыми узлами сети ARPANET, находящимися на расстоянии в 640 км. (в Калифорнийском университете Лос-Анджелеса (*UCLA*) и в Стэнфордском исследовательском институте (*SRI*)) провели сеанс связи. Чарли Клайн пытался выполнить удаленное подключение к компьютеру в *SRI*. Успешную передачу каждого введенного символа его коллега Билл Дювалль из *SRI* подтверждал по телефону. В первый раз удалось отправить всего три символа "LOG", после чего сеть перестала функционировать. LOG должно было быть словом LOGON (команда входа в систему). В рабочее состояние систему вернули уже к 22:30 и следующая попытка оказалась успешной. Именно эту дату можно считать днем рождения *Интернета*.

**1971 год.** Рэй Томлисон, программист из "Bolt Beranek and Newman", разрабатывает систему электронной почты и предлагает использовать значок @.

**1973 год.** Через трансатлантический телефонный кабель к сети были подключены первые иностранные организации из Великобритании и Норвегии.

**1974 год.** Открыта первая коммерческая версия ARPANET – сеть Telenet.

**1975 год.** Джоном Витталом, программистом из университета южной Калифорнии, разработана первая современная почтовая программа, имеющая функциональность "Ответить" и "Переслать".

**1976 год.** Роберт Меткалф, сотрудник исследовательской лаборатории компании Хегох, создает Ethernet – первую локальную компьютерную сеть.

**1977 год.** Число хостов достигло ста. Деннис Хейс и Дейл Хезерингтон разработали первый компьютерный модем.

**1978 год.** Разработана первая доска объявлений (*BBS*). 1978 год также является годом, который принес первое нежелательное коммерческое

сообщение по электронной почте, которое было разослано 600-ам пользователям калифорнийского Arpanet Гарри Зарком.

**1979 год.** Разработана первая многопользовательская игра *MUD* (сокращенно от "многопользовательский лабиринт").

**1980 год.** Европейская организация по ядерным исследованиям *CERN* запустила *ENQUIRE* (написанную Тимом Бернерсом-Ли) – первую гипертекстовую программу.

**1982 год.** Рождение современного *Интернета* – *ARPA* создала единый сетевой язык *TCP/IP*. Активную роль в разработке и стандартизации сетевых протоколов играл Джон Постел.

**1983 год.** 1 января 1983 года сеть *ARPANET* перешла с протокола *NCP* на *TCP/IP*, что позволило разделить эту сеть на *MILNET*, собственно сеть для военных нужд, и *ARPANET*, использовавшуюся в исследовательских целях.

**1984 год.** Число хостов превысило тысячу. Разработана система доменных имен ( *Domain Name System, DNS* ). *DNS* позволила создать масштабируемый распределенный механизм для отображения иерархических имен компьютеров в *Интернет* -адресах. В этом же году в университете Висконсии был создан сервер доменных имен (*Domain Name Server, DNS* ).

Также в 1984 году у сети *ARPANET* появился серьезный соперник: Национальный научный фонд США (*NSF*) основал обширную межуниверситетскую сеть *NSFNet* (*National Science Foundation Network*), которая была составлена из более мелких сетей (включая известные тогда сети *Usenet* и *Bitnet*) и имела гораздо большую пропускную способность, чем *ARPANET*. К этой сети за год подключились около 10 тыс. компьютеров.

**1985 год.** Стюарт Брэнд и Ларри Бриллиант разработали *WELL* (*Whole Earth Lectronic Link*), одно из старейших виртуальных сообществ.

**1988 год.** Разработан протокол *Internet Relay Chat* (*IRC*), благодаря чему в *Интернете* стало возможно общение в реальном времени (чат). Запущен один из первых крупных *Интернет* червей "Червь Морриса", написанный Робертом Моррисом Таппан и вызвавший серьезные перебои в больших частях *Интернета*.

**1989 год.** Число хостов превысило 10 тысяч. В *CERN* родилась концепция Всемирной паутины, предложенная британским ученым Тимом Бернерсом-

Ли. Он же в течение двух лет разработал протокол *HTTP*, язык *HTML* и идентификаторы *URI*.

**1990 год.** В 1990 году сеть ARPANET прекратила свое существование, полностью проиграв конкуренцию *NSFNet*. В том же году было зафиксировано первое подключение к *Интернету* по телефонной линии (*Dialup access*).

**1991 год.** *CERN* создала протокол *World Wide Web* (WWW). Компанией NCR Corporation/AT&T создан Wi-Fi.

**1993 год.** Число *Интернет* -хостов превысило 2 млн., в Сети действует 600 сайтов. Марком Андреесоном в Университете штата Иллинойс создан первый общедоступный графический *Интернет* -браузер *Mosaic*.

**1994 год.** Образовался консорциум W3C (W3 Consortium), который объединил ученых из разных университетов и компаний (в том числе Netscape и Microsoft). С этого времени комитет стал заниматься всеми стандартами в мире *Интернета*.

**1995 год.** *NSFNet* вернулась к роли исследовательской сети, маршрутизацией всего трафика *Интернета* теперь занимались сетевые провайдеры, а не суперкомпьютеры Национального научного фонда. Java и *JavaScript* (первоначально назван LiveScript его создателем, Бренданом Айхом, и включен в состав браузера Netscape Navigator) были впервые представлены публике. Консорциум W3C разработал спецификацию *HTML 2.0*. В данной версии появилась возможность передачи информации с компьютера пользователя на сервер с помощью форм.

**1996 год.** В мире существует 12.8 млн. хостов и 500 тыс. сайтов. Началось соревнование между браузерами Netscape, созданным под руководством Марка Андреесона, и Internet Explorer, разработанным компанией Microsoft. Была запущена первая веб-служба электронной почты – HotMail.

**1997 год.** Начал использоваться термин "блог". В январе 1997 г. W3C создал и принял *HTML 3.2*. Впервые была введена система CSS (Cascading Style Sheets). CSS позволяет осуществить форматирование текста без нарушения логической и структурной разметки. А уже в декабре 1997 г. W3C принимает стандарт *HTML 4.0*, в котором идет разделение на логические и визуальные теги.

**1998 год.** Основана компания Google.

**1999 год.** Впервые предпринята попытка цензуры *Интернета*. В ряде стран государственными органами предприняты серьезные усилия, чтобы технически блокировать доступ пользователей к определенным серверам и сайтам.

**2001 год.** Запущена Wikipedia, по объему сведений и тематическому охвату считающаяся сейчас самой полной энциклопедией из когда-либо создававшихся за всю историю человечества.

**2002 год.** Сеть *Интернет* связывает 689 млн. человек и 172 млн. хостов.

**2003 год.** Создан *Skype*, предоставляющий возможность голосовой связи между компьютерами (VoIP) через *Интернет*.

**2004 год.** Открыт *Facebook*, по состоянию на 2010 год, насчитывающий свыше 400 миллионов активных участников.

**2005 год.** Запущен YouTube.

**2006 год.** Запущен *Twitter*.

**2007 год.** Появился iPhone, который почти полностью отвечает за повышенный интерес к мобильным веб-приложениям.

**2008 год.** Число пользователей, регулярно использующих *Интернет*, составило около 1,5 млрд. человек (около четверти населения Земли).

**2010 год.** Прямой доступ в *Интернет* получил экипаж Международной космической станции.

### **1.1.3. Принципы работы**

Различают логическую и физическую модели *Интернета*. Под логической, прежде всего, понимают Всемирную паутину ( *World Wide Web* ), а под физической – компьютеры, серверы и средства передачи данных между ними.

#### **1.1.3.1. Физическая модель**

Чтобы различные компьютеры сети могли взаимодействовать, они должны использовать один и тот же протокол. Систему протоколов *Интернет* называют "стеком протоколов *TCP/IP* ".

*Стек TCP/IP* объясняется следующими его свойствами [17]:

- Это наиболее завершённый стандартный и в то же время популярный стек сетевых протоколов, имеющий многолетнюю историю.
- Почти все большие сети передают основную часть своего трафика с помощью протокола *TCP/IP*.
- Это метод получения доступа к сети *Интернет*.
- Этот стек служит основой для создания intranet-корпоративной сети, использующей транспортные услуги *Интернет* и гипертекстовую технологию WWW.
- Все современные операционные системы поддерживают стек *TCP/IP*.
- Это устойчивая масштабируемая межплатформенная среда для клиент-серверных приложений.

Так как стек *TCP/IP* был разработан до появления модели взаимодействия открытых систем ISO/OSI, то соответствие уровней стека *TCP/IP* уровням модели **OSI** достаточно условно.

В табл. 1.1 показано, как протоколы *TCP/IP* можно вписать в модель OSI [18]:

Таблица 1.1. Протоколы TCP/IP в модели OSI		
7	Прикладной	<i>HTTP, SMTP, SNMP, FTP, Telnet, scp, SMB, NFS, RTSP, BGP</i>
6	Представительский	<i>XDR, ASN.1, AFP</i>
5	Сеансовый	<i>TLS, SSL, ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP</i>
4	Транспортный	<i>TCP, UDP, RTP, SCTP, SPX, ATP, DCCP, GRE</i>
3	Сетевой	<i>IP, ICMP, IGMP, CLNP, OSPF, RIP, IPX, DDP</i>
2	Канальный	<i>Ethernet, Token ring, PPP, HDLC, X.25, Frame relay, ISDN, ATM, MPLS, Wi-Fi, ARP, RARP</i>
1	Физический	электрические провода, радиосвязь, оптоволоконные провода

На рис. 1.1 показано как 4 уровня протокола *TCP/IP* можно сравнить с уровнями модели OSI [17, 19].



OSI		TCP/IP
7	WWW, Gopher, WAIS, SNMP, FTP telnet, SMTP, TFTP	I
6		
5	TCP, UDP	II
4		
3	IP, ICMP, RIP, OSPF, ARP	III
2	Не регламентируется Ethernet, Token Ring, FDDI, X.25, SLIP, PPP	IV
1		

**Рис. 1.1.** Сравнение модели OSI и стека TCP/IP

Самый нижний (уровень IV, можно условно его назвать "Физический") соответствует физическому и каналному уровням модели OSI. Этот уровень в протоколах *TCP/IP* не регламентируется, но поддерживает все популярные стандарты физического и канального уровня.

Следующий уровень (уровень III, можно условно его назвать "Сетевой") – это уровень межсетевого взаимодействия, который занимается передачей пакетов с использованием различных транспортных технологий локальных сетей, *территориальных сетей*, линий специальной связи и т. п.

Следующий уровень (уровень II, можно условно его назвать "Транспортный") называется основным. На этом уровне функционируют протокол управления передачей **TCP** (Transmission Control Protocol) и протокол дейтаграмм пользователя *UDP* (User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования виртуальных соединений. Протокол *UDP* обеспечивает передачу прикладных пакетов дейтаграммным способом, как и **IP**, и выполняет только функции связующего звена между сетевым протоколом и многочисленными прикладными процессами.

Верхний уровень (уровень I, можно условно его назвать "Прикладной"). К нему относятся такие широко используемые протоколы, как протокол копирования файлов *FTP* (File Transfer Protocol), протокол эмуляции терминала telnet, почтовый протокол *SMTP* (Simple Mail Transfer Protocol),

используемый в электронной почте сети *Интернет*, гипертекстовые сервисы доступа к удаленной информации, такие как WWW и многие другие.

Есть еще целый ряд протоколов, еще не стандартизированных, но уже очень популярных в *Интернете*:

- OSCAR;
- CDDDB;
- MFTP (сеть eDonkey2000);
- BitTorrent;
- Gnutella;
- Skype.

Эти протоколы в большинстве своем нужны для обмена файлами и текстовыми сообщениями, на некоторых из них построены целые файлообменные сети.

Каждый компьютер в сети *TCP/IP* имеет адреса трех уровней:

- *Локальный адрес* узла, определяемый технологией, с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети – это *MAC-адрес* сетевого адаптера или порта маршрутизатора, например, 11-A0-17-3D-BC-01.
- *IP-адрес*, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов или автоматически с помощью протокола *DHCP*(Dynamic Host Configuration Protocol).
- Символьный идентификатор-имя, например, EXAMPLE.COM. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также *DNS* -именем, используется на прикладном уровне, например, в протоколах *FTP* или telnet.

### 1.1.3.2. Логическая модель

**Всемирная паутина** ( *World Wide Web*, Веб) [20] – распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к *Интернету*.

Всемирная паутина основывается на архитектуре клиент-сервер [21]. И на сервере, и на клиенте должно быть установлено дополнительное программное обеспечение – Веб-сервер и Веб-обозреватель соответственно. Это программное обеспечение поддерживает стандартный протокол передачи гипертекстов ( *HTTP* ), а также ряд других протоколов, в частности, протокол передачи файлов ( *FTP* ). Архитектура клиент-сервер означает, что взаимодействие пользовательского компьютера с *Интернетом* происходит следующим образом:

- пользователь вводит адрес ( *URI* или *URL* ) Веб-документа, который он хочет просмотреть;
- Веб-обозреватель формирует соответствующий *HTTP* -запрос к Веб-серверу;
- Веб-сервер находит в Сети запрошенный документ и передает его обозревателю в качестве отклика на запрос (в конечном счете, Веб-сервер находит соответствующий файл на локальном жестком диске и отправляет его по сети запросившему компьютеру);
- обозреватель интерпретирует полученный документ и отображает его пользователю (отображает гипертекст).

Для идентификации ресурсов (зачастую файлов или их частей) во Всемирной паутине используются единообразные идентификаторы ресурсов *URI* (англ. Uniform Resource Identifier). Для определения местонахождения ресурсов в сети используются единообразные локаторы ресурсов *URL* (англ. Uniform Resource Locator). Такие *URL* -локаторы сочетают в себе технологию идентификации *URI* и систему доменных имен *DNS* (англ. *Domain Name System* ) – доменное имя (или непосредственно *IP-адрес* в числовой записи) входит в состав *URL* для обозначения компьютера (точнее – одного из его сетевых интерфейсов), который исполняет код нужного веб-сервера.

Всемирная паутина неразрывно связана с понятиями гипертекста и гиперссылки. Большая часть информации в Веб представляет собой именно гипертекст. Для облегчения создания, хранения и отображения гипертекста во Всемирной паутине традиционно используется язык *HTML* (HyperText

Markup Language), язык разметки гипертекста. После *HTML* -разметки получившийся гипертекст помещается в файл. После того, как *HTML* -файл становится доступен веб-серверу, его начинают называть "веб-страницей". Набор веб-страниц образует веб-сайт. В гипертекст веб-страниц добавляются гиперссылки. Гиперссылки помогают пользователям Всемирной паутины легко перемещаться между ресурсами (файлами) вне зависимости от того, находятся ресурсы на локальном компьютере или на удаленном сервере. Гиперссылки в Веб основаны на технологии *URL*.

В целом можно заключить, что Всемирная паутина стоит на "трех китах" [22, 23]:

- **язык гипертекстовой разметки документов *HTML*** (HyperText Markup Language);
- **универсальный способ адресации ресурсов в сети *URL*** (Universal Resource Locator);
- **протокол обмена гипертекстовой информацией *HTTP*** (HyperText Transfer Protocol).

В последнее время *HTML* начал несколько сдавать свои позиции и уступать их более современным технологиям разметки: *XHTML* и *XML* [20]. *XML* (eXtensible Markup Language) позиционируется как фундамент для других языков разметки. Для улучшения визуального восприятия Веба стала широко применяться технология *CSS*, которая позволяет задавать единые стили оформления для множества веб-страниц.

Популярная концепция развития сети *Интернет* – создание *семантической паутины*. *Семантическая паутина* – это надстройка над существующей Сетью, которая призвана сделать размещенную в ней информацию более понятной для компьютеров. О *семантической паутине* подробнее будет рассказано в "Место веб-разработчика в команде MSF" .

#### **1.1.4. Ключевые термины**

*Интернет, Всемирная паутина, TCP/IP, FTP, SMTP, UDP, MAC-адрес, IP-адрес, DHCP, DNS, HTML, URI, URL, HTTP, Семантическая паутина, XML.*

### **1.2. Эволюция концепции Веб**

#### **1.2.1. Веб 1.0**

Самой простой формулировкой концепции *Веб 1.0* ( *Web 1.0* ) скорее всего, следует считать "тот Веб, который был до *Веб 2.0* " [24].

Переход от *Веб 1.0* к *Веб 2.0* является прямым результатом изменений в поведении тех, кто использует Всемирную Паутину. Основные тенденции *Веб 1.0* включали заботы о проблемах безопасности и приватности в одностороннем потоке информации, через веб-сайты, содержащие материал "только для чтения". Характерным для *Веб 1.0* также являлись компьютерная неграмотность широких масс и распространенность медленных типов подключения к *Интернету*, вдобавок к ограничениям самого *Интернета* [25].

Типичные принципы *Веб 1.0* [24, 26]:

- статичные страницы вместо генерируемого пользователями динамического контента;
- бедная гипертекстовая разметка;
- использование фреймов;
- использование специфичных тегов *HTML* ;
- гостевые книги, форумы или чаты;
- указание конкретного разрешения монитора, при котором дизайн сайта отображается корректно;
- крайне редкое и непопулярное использование стилей *CSS* при оформлении страниц сайта.

Эти принципы описывают лишь общую тенденцию в *Веб 1.0*. Пожалуй, лучше всего сформулировать список принципов *Веб 1.0* можно на основании списка того, с чем боролись последователи *Веб 2.0*.

### **1.2.2. Веб 2.0**

Появление термина *Веб 2.0* принято связывать со статьей "Tim O'Reilly – What Is Web 2.0 " от 30 сентября 2005 года [27]. В этой статье Тим О'Рейли увязал появление большого числа сайтов, объединенных некоторыми общими принципами, с общей тенденцией развития *Интернет* -сообщества, и назвал это явление *Веб 2.0*, в противовес устаревшему *Веб 1.0*.

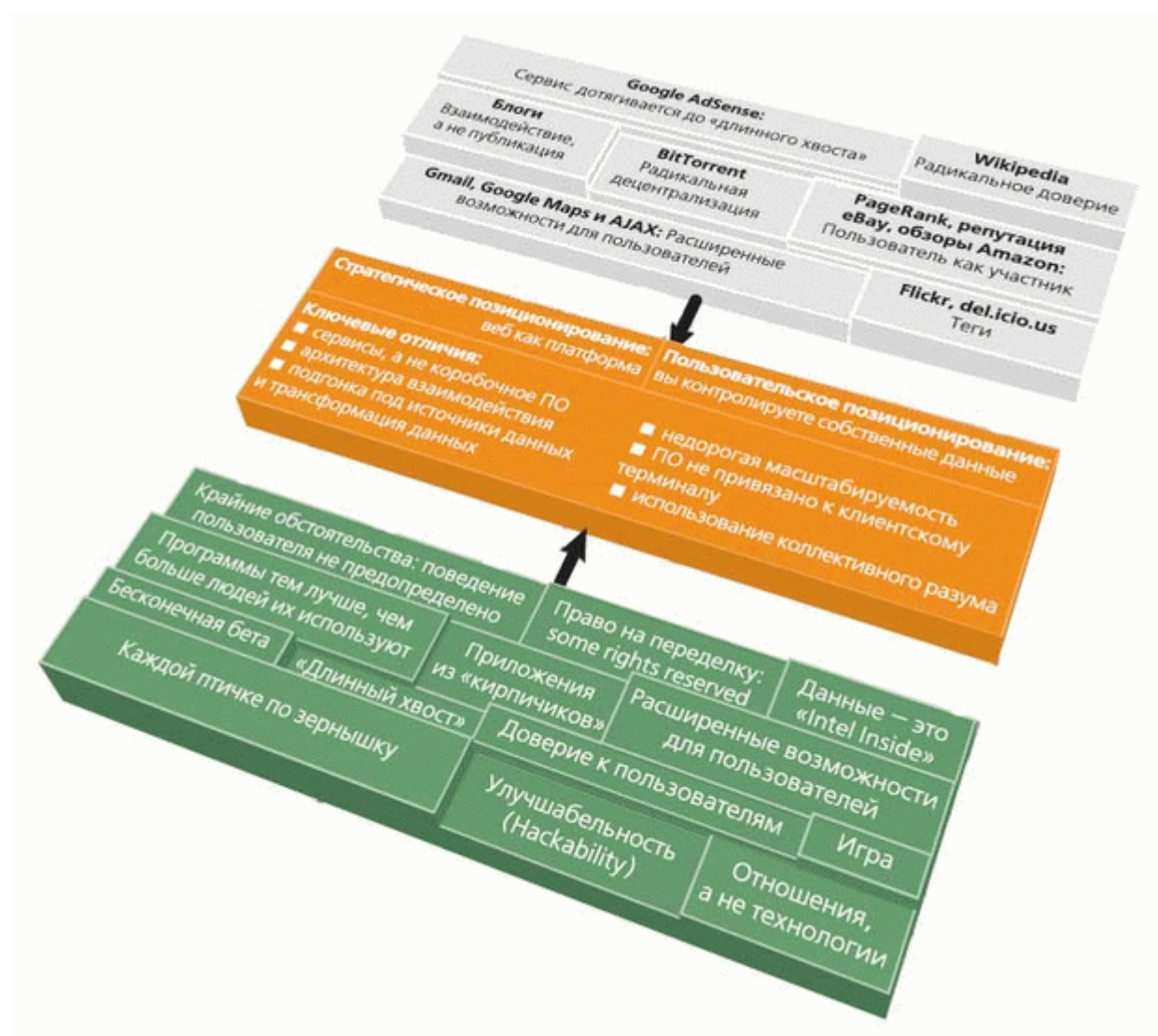
Тим Бернерс-Ли, возглавляющий с 2006 года крупнейший мировой исследовательский проект по изучению всемирной паутины, назвал термин *Web 2.0* простым жаргоном:

"Никто не знает, что это означает. Если *Веб 2.0* – это ваши блоги и вики, тогда это значит "пользователи для пользователей". Но это тоже самое, что сказать – Web существует, чтобы все люди были вместе."

Тим О'Рейли определил *Веб 2.0*, отталкиваясь от конкретных примеров (табл. 1.2) [27]:

Таблица 1.2. Сравнение <i>Веб 1.0</i> и <i>Веб 2.0</i>	
<b><i>Веб 1.0</i></b>	<b><i>Веб 2.0</i></b>
Doubleclick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
Персональные сайты	Блоги
Evite	upcoming.org и EVDB
Спекуляция доменными именами	<i>Поисковая оптимизация</i>
Оплата рекламы по количеству показов	Оплата рекламы по количеству переходов
Извлечение данных из <i>HTML</i>	Веб-сервисы
Публикация	Соавторство
Системы управления контентом (CMS)	Wiki
Каталоги (таксономия)	Теги (фолксономия)
Удержание пользователей	Синдикация контента

На рис. 1.2 показана технологическая карта Веб 2.0, созданная во время мозгового штурма на конференции FOO Camp.



**Рис. 1.2.** Технологическая карта Веб 2.0

*Источник:* Что такое Веб 2.0 [27]

Понятие Веб 2.0 также отразилось и в дизайне [28]. Предпочтительными стали округлость, имитация выпуклых поверхностей, имитация отражений на манер глянцевого пластика современных hi-end устройств (к примеру, плееры). В целом, восприятие внешнего вида на глаз кажется более приятным. Графика таких сайтов занимает больший объем, нежели при использовании аскетичного дизайна. Отчасти эта тенденция связана с совпавшим по времени выходом новых версий операционных систем использующих вышеупомянутые идеи.

Фактически сайт эпохи Веб 2.0 на первый взгляд интерактивен и дружелюбен, позволяет себя легко настраивать. Однако сбор статистики о

пользователях, их предпочтениях и интересах, личной жизни, карьере, круге друзей могут помочь владельцу сайта манипулировать сообществом. По самым пессимистичным прогнозам многочисленные сайты *Веб 2.0* вкупе с другими современными технологиями дают прообраз тоталитарной системы "Большого брата".

### 1.2.3. Веб 3.0

Наиболее распространенной версией трактовки термина *Веб 3.0* является идентификация его как *Семантической Паутины* (Semantic Web) [29]. Главная мысль этой концепции базируется на внедрении мета-языка, описывающего содержание сайтов для организации автоматического обмена между серверами.

*Семантическая паутина* (Semantic Web) – часть глобальной концепции развития сети *Интернет*, целью которой является реализация возможности машинной обработки информации, доступной во Всемирной паутине [30]. Основной акцент концепции делается на работе с метаданными, однозначно характеризующими свойства и содержание ресурсов Всемирной паутины, вместо используемого в настоящее время текстового анализа документов. Термин впервые введен Тимом Бернерсом-Ли в мае 2001 года в журнале "*Scientific American*" [31], и называется им "следующим шагом в развитии Всемирной паутины". В *семантической паутине* предполагается повсеместное использование, во-первых, универсальных идентификаторов ресурсов ( *URI* ), а во-вторых – онтологий и языков описания метаданных.

Эта концепция была принята и продвигается Консорциумом W3C [32]. Для ее внедрения предполагается создание сети документов, содержащих метаданные о ресурсах Всемирной паутины и существующей параллельно с ними. Тогда как сами ресурсы предназначены для восприятия человеком, метаданные используются машинами (*поисковыми роботами* и другими интеллектуальными агентами) для проведения однозначных логических заключений о свойствах этих ресурсов.

10 февраля 2004 г. на сайте W3C появляется описание языка "*OWL*" (язык описания онтологий).

Через полгода новый язык описания онтологий *OWL* стал поддерживать редактор онтологий Protege – разработка Стэнфордского Университета. В это же время Semantic Web начало активно интересоваться международное



научное сообщество. В разных изданиях появляется множество статей по Semantic Web.

В 2005 г. на сайте W3 появляется описание RDF/A – синтаксиса, который уже сейчас позволяет встраивать метаданные RDF в документы XHTML.

10 марта 2006 г. выходит RDF/A Primer. Таким образом, Semantic Web был "привязан" к XHTML.

В 2006 г. также завершилась разработка языка запросов к RDF документам с SQL-подобным синтаксисом, его окончательное название – *SPARQL*.

Некоторые авторы дают другую трактовку термину *Веб 3.0*. Так Джейсон Калаканис определяет *Веб 3.0* [33] как высококачественный контент и сервисы, которые создаются талантливыми профессионалами на технологической платформе *Веб 2.0*.

По сути, *Веб 3.0* использует технологическую базу *Веб 2.0* [29, 30]:

- *AJAX* – загрузка данных без перезагрузки самой веб-страницы;
- *RIA (Adobe Flex, JavaFX, Microsoft Silverlight )*;
- *XML (eXtensible Markup Language)* – язык разметки данных, представляющий собой свод общих синтаксических правил;
- *RSS (Really Simple Syndication)* – семейство *XML* -форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т. п.;
- Теги – отображение тегов в виде облака, что значительно упрощает определение пользователем наиболее актуальной информации;
- Блоговая структура информации – ленточная подача информации, где поток идет по убыванию сверху-вниз, а метод сортировки задает пользователь.

Главная идея *Веб 3.0*, по мнению этих авторов, состоит в том, чтобы пользователь, который до этого единолично был вовлечен в процесс формирования контента, отныне творит коллективно и его партнерами, помимо других пользователей, являлись эксперты различных направлений, причем статус пользователя может быть изменен на экспертный, равно, как и форма сотрудничества создателя контента и портала. Эксперт должен выступить своеобразным модератором публикуемого контента. По сути, не

исключается и возможность платной основы для сотрудничества, но гораздо более важным моментом является появление в порталах формата *Веб 3.0* "коллективного разума" (*wisdom of the crowds*), вместо господствующего сегодня "группового сумасшествия" (*madness of the mobs*). *Веб 3.0* предполагает появление узкоспециализированных ресурсов, где будет произведена агрегация всех необходимых пользователю сервисов и инструментов профессиональной социальной составляющей и будет осуществляться публикация экспертно-модерируемого контента.

#### **1.2.4. Выводы**

Обобщив все вышесказанное, можно выделить общие признаки Веб-концепций [34, 35].

##### *Веб 1.0 – Интернет как информационный портал*

- эксклюзивность информации, необходимо быть первым собственником контента;
- разделение *World Wide Web* на пригодные для использования каталоги;
- каждый человек имеет свой собственный личный уголок в киберпространстве;
- недостатки:
  - Контекст;
  - Взаимодействие;
  - Масштабируемость;
- Примеры сайтов:
  - Ofoto;
  - Hotmail;
  - Dmoz;
  - GeoCities.

##### *Веб 2.0 – Сеть как платформа*

- фокус на сообществах для создания и проверки контента;

- свободная форма организации и классификации контента посредством тэгов;
- создание "интерфейсов" для будущей интеграции (RSS, API);
- недостатки:
  - Персонализация;
  - Мобильность и портативность;
  - Совместимость;
- Примеры сайтов:
  - YouTube;
  - Flickr;
  - *Facebook*.

### *Веб 3.0 – Интернет как экспертная система*

- извлечение проверенной информации;
- мобильность и портативность;
- повсеместное использование RSS и API;
- "дайте мне то, что, как вы думаете, я хочу на основе того, где я был и что делал";
- Примеры сайтов:
  - Google : универсальный поиск + история пользовательского поиска;
  - *FOAF* (friend of a friend) : моя жизнь в RDF;
  - Wink : социальная поисковая система;
  - *Twitter* : "король" микроблогов;
  - Surface : технология multi-touch для сенсорных экранов;
  - OpenID : единая авторизация в *Интернете*.

Различия концепций также можно представить следующей таблицей (табл. 1.3 [35]):

Таблица 1.3. Сравнение концепций <i>Веб 1.0</i> , <i>Веб 2.0</i> , <i>Веб 3.0</i>			
Концепция Свойство	<i>Web 1.0</i>	<i>Web 2.0</i>	<i>Web 3.0</i>
<i>Концепция</i>	Веб только для чтения ("the mostly read only web")	Веб для бурного чтения-записи ("the wildly read-write web")	Портативный индивидуальный Веб ("the portable personal web")
<i>Количество пользователей</i>	45 миллионов глобальных пользователей (1996)	Большее 1 миллиарда глобальных пользователей (2006)	Еще больше
<i>Ориентация</i>	Ориентация на компании (focused on companies)	Ориентация на сообщества (focused on communities)	Ориентация на индивидуальностях (focused on the individual)
<i>Структура данных</i>	Домашние страницы (home pages)	Блоги (blogs)	Lifestreaming-функции (lifestream)
<i>Концепция данных</i>	Владение контентом (owning content)	Обмен контентом (sharing content)	Объединение динамического контента ( <i>consolidating dynamic content</i> )
<i>Управление знаниями</i>	Britannica Online	Wikipedia	<i>Интернет</i>
<i>Технологии</i>	<i>HTML</i> , порталы	<i>XML</i> , RSS	Технологии "drag and drop" и mashups
<i>Представление</i>	Веб формы	Веб приложения	Виджеты (widgets) и гаджеты ( <i>gadgets</i> )
<i>Классификация</i>	Директории (иерархическое строение)	Тэги (практика совместной категоризации)	Поведение пользователей (большая



## 1.3. Основные технологии в Интернете

### 1.3.1. Веб-программирование

**Веб-программирование** (Веб-разработка) – это бурно развивающийся раздел программирования, ориентированный на разработку динамических *Интернет-приложений* [37].

Языки *веб-программирования* делятся на две группы: клиентские и серверные.

**Клиентские языки** обрабатываются на стороне пользователя (в основном в браузере). Соответственно обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так, чтобы тот вообще игнорировал скрипты. При этом если браузер старый, он может не поддерживать тот или иной язык или версию языка, на которую опирался разработчик. С современными браузерами таких проблем возникать не должно, к тому же языки программирования не так уж часто кардинально обновляются (раз в несколько лет) и лучшие из них давно известны. Код клиентского скрипта может посмотреть каждый, выбрав в меню своего браузера "Исходный код страницы".

Преимущество клиентского языка заключается в том, что обработка скриптов на таком языке может выполняться без отправки документа на сервер. Программа сразу проверит правильное заполнение формы перед отправкой, и, если необходимо, выведет ошибку. Отсюда же вытекает и то ограничение, что с помощью клиентского языка программирования ничто не может быть записано на сервер.

Самым распространенным из клиентских языков является *JavaScript*, разработчиками которого является компания Netscape совместно с компанией Sun Microsystems. Еще один популярный язык – это *VBScript*. Помимо этого в последнее время набрали популярность такие технологии как *AJAX*, *Adobe Flash*, *Microsoft Silverlight* и др.

**Серверные языки программирования** открывают перед программистом большие просторы в деятельности.

Когда пользователь делает запрос на какую-либо страницу (переходит на нее по ссылке, или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере (то есть выполняются все программы, связанные со страницей) и только потом возвращается к

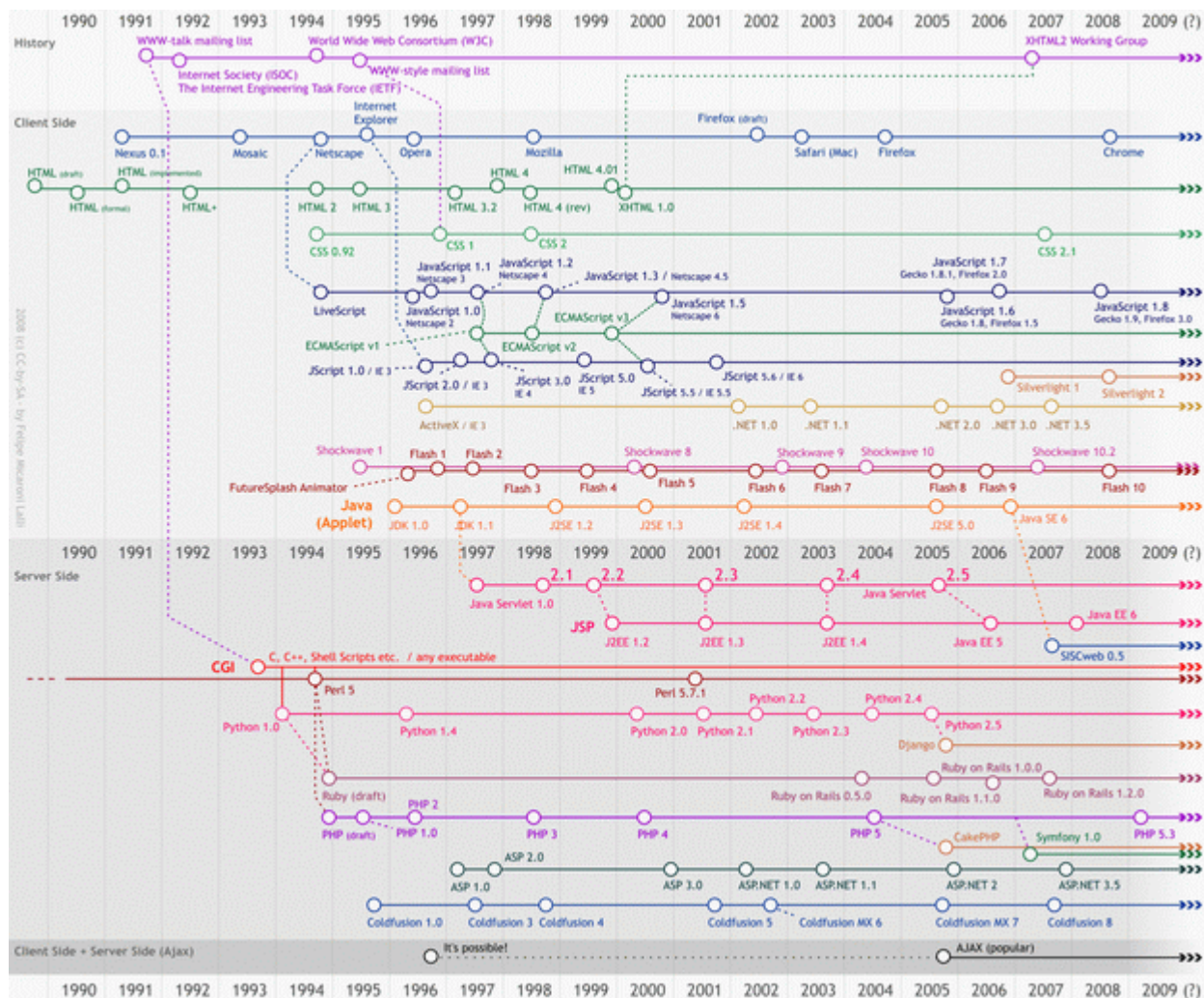
посетителю по сети в виде файла. Этот файл может иметь расширения: *HTML, PHP, ASP, Perl, SSI, XML, DHTML, XHTML*.

Работа программ уже полностью зависима от сервера, на котором расположен сайт, и от того, какая версия того или иного языка поддерживается.

Важной стороной работы серверных языков является Система управления базами данных (СУБД). Это, по сути, тоже сервер, на котором в определенном пользователем порядке хранится разная необходимая информация, которая может быть вызвана в любой момент. Популярными среди систем управления базами данных являются:

- IBM DB2;
- Microsoft SQL Server;
- MySQL;
- Oracle;
- PostgreSQL;
- SQLite.

Хронология развития веб-технологий показана на рис. 1.4.



**Рис. 1.4.** Хронология развития веб-технологий

Источник: Wikipedia [37]

Рассмотрим подробнее наиболее известные языки и технологии.

### 1.3.2. HTML

*HTML* (HyperText Markup Language – "язык разметки гипертекста") – стандартный язык разметки документов во Всемирной паутине [38]. Большинство веб-страниц создаются при помощи языка *HTML*. Язык *HTML* интерпретируется браузером и отображается в виде документа, в удобной для человека форме. *HTML* является приложением SGML (стандартного обобщенного языка разметки) и соответствует международному стандарту ISO 8879.

Язык *HTML* был разработан британским ученым Тимом Бернерсом-Ли приблизительно в 1991-1992 годах в стенах Европейского совета по ядерным исследованиям в Женеве (*CERN*). *HTML* создавался как язык для обмена



научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области верстки. *HTML* успешно справлялся с проблемой сложности *SGML* путем определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют "тегами". С помощью *HTML* можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в *HTML* внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже. Изначально язык *HTML* был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения).

Текстовые документы, содержащие код на языке *HTML* (такие документы традиционно имеют расширение .html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые "браузерами" или "*Интернет* -обозревателями", обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введенных пользователем данных на сервер.

Наиболее популярными на сегодняшний день браузерами являются (по данным на январь 2010 г.) [39]:

- Internet Explorer – 62,12%;
- Mozilla Firefox – 24,43%;
- Google Chrome – 5,22%;
- Safari – 4,53%;
- Opera – 2,38%.

Спецификация *HTML* имеет несколько версий:

- RFC 1866 – *HTML* 2.0, одобренный как стандарт 22 сентября 1995 года;
- *HTML* 3.2 [36] – 14 января 1997 года;
- *HTML* 4.0 [37] – 18 декабря 1997 года;
- *HTML* 4.01 [38] – 24 декабря 1999 года;

- ISO/IEC 15445:2000 [39] (так называемый ISO *HTML*, основан на *HTML* 4.01 Strict) – 15 мая 2000 года.

Подробнее данная технология будет освещена в "Создание статического содержания. HTML" .

### 1.3.3. XHTML

**XHTML** (Extensible Hypertext Markup Language – расширяемый язык разметки гипертекста) – язык разметки веб-страниц, по возможностям сопоставимый с *HTML*, созданный на базе *XML* [40]. Как и *HTML*, XHTML соответствует спецификации SGML, поскольку *XML* является ее подмножеством.

Стандарт XHTML построен не как самодостаточное описание языка, а как перечень различий между *HTML* 4.01 и XHTML [41]:

- Все элементы должны быть закрыты. Теги, которые не имеют закрывающего тега (например, `<img>` или `<br>` ) должны иметь на конце / (например, `<br />` ).
- Булевы атрибуты записываются в развернутой форме. Например, следует писать `<option selected="selected">` или `<td nowrap="nowrap">`.
- Имена тегов и атрибутов должны быть записаны строчными буквами (например, `<img alt="" />` вместо `<IMG ALT="" />` ).
- XHTML гораздо строже относится к ошибкам в коде; `<` и `&` везде, даже в *URL*, должны замещаться `&lt;` и `&amp;`; соответственно. По рекомендации W3C браузеры, встретив ошибку в XHTML, должны сообщить о ней и не обрабатывать документ.
- Кодировкой по умолчанию является UTF-8 (в отличие от *HTML*, где кодировкой по умолчанию является ISO 8859-1).

Для XHTML страниц рекомендуется задавать MIME-тип – `application/xhtml+xml`, но это не является обязательным, более того – браузер Internet Explorer 8 и младшие версии, не смогут обрабатывать страницу, поэтому с XHTML 1.0 традиционно используется MIME-тип для HTML – `text/html`.

Версии XHTML:

- XHTML 1.0 Переходный (Transitional): предназначен для легкой миграции из *HTML* 3.2 и для тех, кто использует инлайн-фреймы.
- XHTML 1.0 Строгий (Strict): полностью отделяет содержание документа от оформления (задается только через *CSS* ), многие атрибуты (такие как, например, *bcolor* и *align*) более не поддерживаются, их поведение можно задавать только через таблицу стилей.
- XHTML 1.0 Фреймовый (Frameset): используется, если необходимо разделить окно браузера на несколько фреймов.
- XHTML 1.1 Модульный (Module-based): авторы могут импортировать дополнительные свойства в их разметку.
- XHTML Основной (Basic): специальная облегченная версия XHTML для устройств, которые не могут использовать полный набор элементов XHTML – в основном используется в миниатюрных устройствах, таких как мобильные телефоны. Подразумевается, что он заменит *WML* и *C-HTML*.
- XHTML мобильного профиля (Mobile Profile): основанный на XHTML Basic, добавляет специфические элементы для мобильных телефонов.
- XHTML 2.0. Пока в разработке. Синтаксис еще больше приближен к синтаксису *XML*. Также является модульным языком.

Подробнее данная технология будет освещена в "Создание статического содержания. HTML" .

#### 1.3.4. CSS

*CSS* (Cascading Style Sheets – каскадные таблицы стилей) – технология описания внешнего вида документа, написанного языком разметки [42]. Преимущественно используется как средство оформления веб-страниц в формате *HTML* и XHTML, но может применяться с любыми видами документов в формате *XML*, включая *SVG* и *XUL*.

Термин "каскадные таблицы стилей" был предложен Хокон Виум Ли в 1994 году. Совместно с Бертом Босом он стал развивать *CSS*.

*CSS* используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа. Основной целью разработки *CSS* являлось разделение содержимого (написанного на *HTML*

или другом языке разметки) и представления документа (написанного на CSS ) [43]. Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана) и др.

Наиболее полно поддерживающими стандарт CSS являются браузеры, работающие на Gecko (Mozilla Firefox и др.) и WebKit (Arora, Google Chrome, Safari), а также браузер Opera [44].

Что касается Internet Explorer, то только 8-ая его версия полностью поддерживает CSS 2.1 и частично – CSS 3 [45].

Преимущества применения CSS [42, 43]:

- несколько дизайнов страницы для разных устройств просмотра;
- уменьшение времени загрузки страниц сайта за счет переноса правил представления данных в отдельный CSS -файл;
- простота последующего изменения дизайна;
- дополнительные возможности оформления, например, с помощью CSS -верстки можно сделать блок текста, который остальной текст будет обтекать или сделать так, чтобы меню было всегда видно при прокрутке страницы.

Недостатки применения CSS [42, 43]:

- различное отображение верстки в различных браузерах (особенно устаревших), которые по-разному интерпретируют одни и те же данные CSS ;
- часто встречающаяся необходимость на практике исправлять не только один CSS -файл, но и теги *HTML* и серверный код, которые сложным и ненаглядным способом связаны с селекторами CSS.

Подробнее данная технология будет освещена в "Создание статического содержания. HTML" .

### 1.3.5. XML

*XML* (eXtensible Markup Language – расширяемый язык разметки) – рекомендованный Консорциумом Всемирной паутины язык разметки, фактически представляющий собой свод общих синтаксических правил [46]. *XML* – текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML), иногда называемых словарями. *XML* является упрощенным подмножеством языка SGML. Годом рождения *XML* можно считать 1996 год, в конце которого появился черновой вариант спецификации языка, или 1998, когда эта спецификация была утверждена.

Целью создания *XML* было обеспечение совместимости при передаче структурированных данных между разными системами обработки информации, особенно при передаче таких данных через *Интернет*. Словари, основанные на *XML* (например, RDF, RSS, MathML, XHTML, SVG), сами по себе формально описаны, что позволяет программно изменять и проверять документы на основе этих словарей, не зная их семантики, то есть, не зная смыслового значения элементов. Важной особенностью *XML* также является применение так называемых пространств имен (namespace).

К достоинствам использования *XML* можно отнести [46, 47]:

- *XML* – язык разметки, позволяющий отобразить двоичные данные в текст, читаемый человеком и анализируемый компьютером;
- *XML* поддерживает Юникод;
- в формате *XML* могут быть описаны такие структуры данных как записи, списки и деревья;
- *XML* – это самодокументируемый формат, который описывает структуру и имена полей так же как и значения полей;
- *XML* имеет строго определенный синтаксис и требования к анализу, что позволяет ему оставаться простым, эффективным и непротиворечивым;
- *XML* – формат, основанный на международных стандартах;
- Иерархическая структура *XML* подходит для описания практически любых типов документов, кроме аудио и видео мультимедийных

потоков, растровых изображений, сетевых структур данных и двоичных данных;

- *XML* представляет собой простой текст, свободный от лицензирования и каких-либо ограничений;
- *XML* не зависит от платформы;
- *XML* является подмножеством SGML;
- уже накоплен большой опыт работы с языком и созданы специализированные приложения;
- *XML* не накладывает требований на расположение символов в строке;
- в отличие от бинарных форматов, *XML* содержит метаданные об именах, типах и классах описываемых объектов, по которым приложение может обработать документ;
- *XML* имеет реализации парсеров для всех современных языков программирования;
- *XML* поддерживается на низком аппаратном, микропрограммном и программном уровнях в современных аппаратных решениях.

К недостаткам *XML* можно отнести [46, 47, 48]:

- синтаксис *XML* избыточен:
  - размер *XML* документа существенно больше бинарного представления тех же данных;
  - избыточность *XML* может повлиять на эффективность приложения (возрастает стоимость хранения, обработки и передачи данных);
  - для большого количества задач не нужна вся мощь синтаксиса *XML* и можно использовать значительно более простые решения;
- неоднозначность моделирования, т.е. нет общепринятой методологии для моделирования данных в *XML* ;
- *XML* не содержит встроенной в язык поддержки типов данных;

- *иерархическая модель данных*, предлагаемая *XML*, ограничена по сравнению с реляционной моделью и объектно-ориентированными графами и *сетевой моделью данных*;
- пространства имен *XML* сложно использовать и их сложно реализовывать в *XML* -парсерах.

Наиболее распространены три способа преобразования *XML* -документа в отображаемый пользователю вид:

1. применение стилей *CSS* ;
2. применение преобразования *XSLT*;
3. написание на каком-либо языке программирования обработчика *XML* - документа.

Без использования *CSS* или *XSL* *XML* -документ отображается как простой текст в большинстве Веб-браузеров. Некоторые браузеры, такие как Internet Explorer, Mozilla и Mozilla Firefox отображают структуру документа в виде дерева, позволяя сворачивать и разворачивать узлы с помощью нажатий клавиши мыши.

Подробнее данная технология будет освещена в "Разработка на ASP.NET. MasterPage, динамические компоненты, AJAX, ASP.NET MVC".

### **1.3.6. JavaScript**

Объектно-ориентированный скриптовый язык программирования *JavaScript* (первоначально назван LiveScript его создателем, Бренданом Вашем, и развернут в составе браузера Netscape Navigator) был впервые представлен публике в 1995 году [49].

*JavaScript* обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: *динамическая типизация*, слабая типизация, *автоматическое управление памятью*, прототипное программирование, функции как объекты первого класса.

На *JavaScript* оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом легким для использования непрограммистами. Языком *JavaScript* не владеет какая-либо компания или

организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

*JavaScript* обладает рядом свойств объектно-ориентированного языка, но реализованное в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными объектно-ориентированными языками. Кроме того, *JavaScript* имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придает языку дополнительную гибкость.

Структурно *JavaScript* можно представить в виде объединения трех четко различимых друг от друга частей:

- ядро (ECMAScript);
- объектная модель браузера (*Browser Object Model* или *BOM*);
- объектная модель документа (*Document Object Model* или *DOM*).

Если рассматривать *JavaScript* в отличных от браузера окружениях, то объектная модель браузера и объектная модель документа могут не поддерживаться. Объектную модель документа иногда рассматривают как отдельную от *JavaScript* сущность, что согласуется с определением DOM как независимого от языка интерфейса документа.

Для добавления *JavaScript* -кода на страницу, можно использовать теги `<script></script>`.

Область применения *JavaScript* очень широка:

- в клиентской части веб-приложений;
- в AJAX ;
- в технологии *Comet*;
- в браузерных операционных системах;
- для создания небольших программ, размещаемых в закладки браузера (Букмарклеты);
- приложения, написанные на *JavaScript*, могут исполняться на серверах, использующих Java 6 и более поздних версий, что используется для



построения серверных приложений, позволяющих обрабатывать *JavaScript* на стороне сервера;

- в качестве языка разработки мобильных приложений (на платформе Mojo SDK в Palm webOS);
- для реализации виджетов, так и для реализации движков виджетов (Apple\_Dashboard, Microsoft *Gadgets*, Google Desktop *Gadgets*, Klipfolio Dashboard);
- для написания прикладного ПО (57 % исходного кода Mozilla Firefox написано на *JavaScript* );
- в качестве скриптового языка доступа к объектам приложений;
- в офисных приложениях для автоматизации рутинных действий, написания макросов, организации доступа со стороны веб-служб;
- в Excel Services 2010 добавились два новых *интерфейса программирования приложений*: REST API и *JavaScript Object Model (JSOM)*.

Для обеспечения высокого уровня абстракции и достижения приемлемой степени кросс-браузерности при разработке веб-приложений используются библиотеки *JavaScript*. Они представляют собой набор многократно используемых объектов и функций. Среди известных *JavaScript* библиотек можно отметить Adobe life, *Dojo Toolkit*, *Extjs*, *jQuery*, Mootools, Prototype, Qooxdoo.

На сегодняшний день поддержку *JavaScript* обеспечивают современные версии всех наиболее часто используемых браузеров. В Internet Explorer, Mozilla Firefox, Safari, Chrome, Opera имеется полная поддержка третьей редакции ECMA-262.

Подробнее данная технология будет освещена в "Создание динамического наполнения страницы. Основы JavaScript" .

### **1.3.7. PHP**

*PHP* ( *PHP: Hypertext Preprocessor* – " *PHP: препроцессор гипертекста*", Personal Home Page Tools – "Инструменты для создания персональных веб-страниц") – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений [50]. В настоящее время поддерживается подавляющим большинством хостинг-

провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект не является свободным и распространяется под собственной лицензией.

В области программирования для *Интернета PHP* – один из популярнейших скриптовых языков (наряду с JSP, *Perl* и языками, используемыми в *ASP.NET*) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии *PHP*.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений.

В настоящее время *PHP* используется сотнями тысяч разработчиков. Согласно рейтингу Tiobe, базирующемуся на данных поисковых систем, в декабре 2009 года *PHP* находится на 3 месте среди языков программирования (уступая Java и C), поднявшись за год на две позиции [51]. К крупнейшим сайтам, использующим *PHP*, относятся *Facebook* (который, однако, использует транслятор кода HipHop с *PHP* на C++ с целью оптимизации), ВКонтакте, Wikipedia.

В настоящий момент существует единственная реализация *PHP*, ни одна сторонняя компания не поддерживает исполняемых модулей, отличных от официальной сборки. Такое положение вещей, с одной стороны, позволяет быстро внедрять и распространять нововведения среди сообщества разработчиков, с другой стороны, разрабатывать язык программирования в условиях отсутствия стандарта, так как единая реализация обеспечивает его по факту. В таких условиях большое значение приобретает версия интерпретатора, определяющая текущую функциональность (обратная совместимость между версиями интерпретатора не соблюдается строго).

### **1.3.8. Perl**

*Perl* – высокоуровневый интерпретируемый динамический язык программирования общего назначения, созданный в 1987 г. Ларри Уоллом, лингвистом по образованию [52]. Название языка представляет собой аббревиатуру, которая расшифровывается как Practical Extraction and Report Language "практический язык для извлечения данных и составления отчетов".

Согласно Ларри Уоллу, *Perl* имеет два девиза. Первый – "There's more than one way to do it" ("Есть больше одного способа сделать это", также известный как TMTOWTDI); второй – "Easy things should be easy and hard things should be possible" ("Простые вещи должны быть простыми, а сложные вещи – возможными").

Основной особенностью языка считаются его богатые возможности для работы с текстом, в том числе реализованные при помощи регулярных выражений. Перл унаследовал много свойств от языков C, shell script, *awk*.

*Perl* также знаменит огромной коллекцией дополнительных модулей CPAN [53].

### 1.3.9. AJAX

*AJAX* (Asynchronous Javascript and XML – "асинхронный JavaScript и XML") – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в "фоновом" обмене данными браузера с веб-сервером [54]. В результате, при обновлении данных веб-страница не перезагружается полностью и веб-приложения становятся более быстрыми и удобными.

Впервые термин *AJAX* был публично использован в 2005 году в статье Джесси Джеймса Гарретта (Jesse James Garrett) "Новый подход к веб-приложениям" [55]. Гарретт придумал термин, когда ему пришлось как-то назвать новый набор технологий, предлагаемый им клиенту.

*AJAX* – не самостоятельная технология, а концепция использования нескольких смежных технологий. *AJAX* базируется на двух основных принципах:

- использование технологии динамического обращения к серверу "на лету", без перезагрузки всей страницы полностью, например:
  - с использованием XMLHttpRequest (основной объект);
  - через динамическое создание дочерних фреймов;
  - через динамическое создание тега <script>.
- использование DHTML для динамического изменения содержания страницы;

В качестве формата передачи данных обычно используются JSON или XML.

К преимуществам *AJAX* можно отнести:

- использование *AJAX* позволяет значительно сократить трафик при работе с веб-приложением;
- *AJAX* позволяет несколько снизить нагрузку на сервер;
- ускорение реакции интерфейса.

К недостаткам *AJAX* можно отнести:

- отсутствие интеграции со стандартными инструментами браузера;
- динамически загружаемое содержимое обычно недоступно поисковикам;
- старые методы учета статистики сайтов становятся неактуальными.

Подробнее данная технология будет освещена в "Разработка на ASP.NET. Жизненный цикл страницы, пользовательский интерфейс" .

### **1.3.10. Adobe Flash**

*Adobe Flash* (ранее известная как Macromedia Flash) – мультимедийная платформа, используемая для создания векторной анимации и интерактивных приложений, а также для интеграции видеороликов в веб-страницы [56].

Разработка Flash была начата компанией FutureWave, создавшей пакет анимации FutureSplash Animator. В 1996 году FutureWave была приобретена компанией Macromedia, которая переименовала FutureSplash Animator в Flash. Под этим наименованием платформа продолжает развиваться и поныне (хотя после того, как в 2005 году компания Macromedia была поглощена Adobe, Macromedia Flash стал официально называться *Adobe Flash* ).

*Adobe Flash* позволяет работать с векторной, растровой и ограниченно с трехмерной графикой, а также поддерживает двунаправленную потоковую трансляцию аудио и видео.

В основе Flash лежит векторный морфинг, то есть плавное "перетекание" одного ключевого кадра в другой. Это позволяет делать сложные мультипликационные сцены, задавая лишь несколько ключевых кадров для каждого персонажа.

Flash использует язык программирования ActionScript, основанный на ECMAScript.

В качестве основного средства разработки используется пакет *Adobe Flash Professional* (последняя версия *Adobe Flash CS4*).

Стандартным расширением для скомпилированных Flash-файлов является SWF (Shockwave Flash или Small Web Format). Видеоролики в формате Flash представляют собой файлы с расширением FLV (при этом Flash в данном случае используется только как контейнер для видеозаписи).

Flash-контент воспроизводится с помощью целого ряда программных средств, но *доминирующее положение* на рынке занимает официальный *Adobe Flash Player*, распространяемый в качестве бесплатного плагина для большинства современных браузеров.

Основной недостаток Flash-приложений – чрезмерная требовательность к ресурсам процессора. Недостаточная мощность компьютера может повлиять на производительность операционной системы в целом, либо привести к искажению результатов работы Flash-приложения, связанных с отображением анимации или подсчетом времени.

Другой важный недостаток заключается в том, что не всегда есть возможность запустить Flash-приложение, либо она связана с некоторыми трудностями (например, необходимо установить плагин или обновить его до последней версии).

Также использование Flash для размещения текстовой информации препятствует ее индексированию поисковыми системами. Однако существует множество способов решить эту проблему.

Подробнее данная технология будет освещена в "Rich Internet application" и "Безопасность в Веб-разработке".

### **1.3.11. Silverlight**

*Microsoft Silverlight* – это мультимедийная технология схожая по решаемым задачам с *Adobe Flash*. Также *Microsoft Silverlight* – это плагин для браузера, который позволяет запускать приложения, содержащие анимацию, векторную графику и аудио-видео ролики, что характерно для RIA (*Rich Internet application*) [57].

Silverlight реализован для ОС Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7, Mac OS X 10.4, Mac OS X 10.5 и браузеров

Internet Explorer 6.0/7.0/8.0, Mozilla Firefox 1.5/2.0/3, Safari 3.1, Google Chrome 3.0. В будущем также планируется поддержка Opera, мобильных устройств, начиная с Windows Mobile 6 и Symbian (Series 60), и, возможно, других платформ.

Silverlight предоставляет графическую систему, схожую с Windows Presentation Foundation, и объединяет мультимедиа, графику, анимацию и интерактивность в одной программной платформе. Он был разработан, чтобы работать с XAML и с языками Microsoft .NET. XAML используется для разметки страниц, использующих векторную графику и анимацию. Текст, содержащийся в Silverlight приложениях, доступен для поисковых систем, так как он не компилируется, а доступен в виде XAML. Silverlight также можно использовать для того, чтобы создавать виджеты для Windows *SideBar* в Windows Vista.

Silverlight поддерживает воспроизведение WMV, WMA и MP3 для всех поддерживаемых браузеров, не требуя при этом дополнительных компонентов, таких как Windows Media Player.

Silverlight позволяет динамически загружать XML и использовать DOM для взаимодействия с ним так же, как это делается в Ajax. Silverlight содержит объект Downloader, благодаря которому можно скачивать скрипты, медиа файлы и т. д., если это необходимо приложению. Начиная с версии 2.0, логика программы может быть описана в любом из языков .NET, включая динамические языки программирования, такие как Iron Ruby и Iron Python, которые в свою очередь исполняются в DLR (*Dynamic Library Runtime*), а не CLR (*Common Language Runtime*).

Подробнее данная технология будет освещена в "Rich Internet application" и "Безопасность в Веб-разработке".

### **1.3.12. ASP.NET**

ASP.NET – технология создания веб-приложений и веб-сервисов от компании Microsoft [58]. Она является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP.

Хотя ASP.NET берет свое название от старой технологии Microsoft ASP, она значительно от нее отличается. Microsoft полностью перестроила ASP.NET, основываясь на Common Language Runtime (CLR), который является основой всех приложений Microsoft .NET. Разработчики могут писать код для ASP.NET, используя практически любые языки программирования, в том

числе, и входящие в комплект .NET Framework (C#, Visual Basic.NET, и JScript .NET). *ASP.NET* имеет преимущество в скорости по сравнению со скриптовыми технологиями, так как при первом обращении код компилируется и помещается в специальный кэш, и впоследствии только исполняется, не требуя затрат времени на *парсинг*, оптимизацию, и т. д.

Вместе с тем следует учитывать, что указанное преимущество не всегда может быть реализовано. Это связано с тем, что на скорость работы реального проекта влияют множество факторов. В первую очередь это квалификация руководителей разработки и исполнителей: медленные алгоритмы легко сводят на нет незначительное преимущество скомпилированного кода перед интерпретацией серверных скриптов

Преимущества *ASP.NET* перед *ASP*:

- *компилируемый код* выполняется быстрее, большинство ошибок отлавливается еще на стадии разработки;
- значительно улучшенная обработка ошибок времени выполнения, с использованием блоков `try..catch` ;
- пользовательские элементы управления ( `controls` ) позволяют выделять часто используемые шаблоны, такие как меню сайта;
- использование метафор, уже применяющихся в Windows-приложениях, например, таких как элементы управления и события;
- расширяемый набор элементов управления и библиотек классов позволяет быстрее разрабатывать приложения;
- *ASP.NET* опирается на многоязыковые возможности .NET, что позволяет писать код страниц на VB.NET, Delphi.NET, Visual C#, J# и т. д.;
- возможность кэширования всей страницы или ее части для увеличения производительности;
- возможность кэширования данных, используемых на странице;
- возможность разделения визуальной части и бизнес-логики по разным файлам ("code behind");
- расширяемая модель обработки запросов;
- расширенная событийная модель;

- расширяемая модель серверных элементов управления;
- наличие master-страниц для задания шаблонов оформления страниц;
- поддержка *CRUD* операций при работе с таблицами через GridView ;
- встроенная поддержка *AJAX*.

Подробнее данная технология будет освещена в "Разработка на ASP.NET. Жизненный цикл страницы, пользовательский интерфейс" и "Разработка на ASP.NET. MasterPage, динамические компоненты, AJAX, ASP.NET MVC".

### 1.3.13. Ключевые термины

*Веб-программирование*, Клиентские языки программирования, Серверные языки программирования, HTML, XHTML, CSS, XML, JavaScript, PHP, Perl, AJAX, Adobe Flash, Microsoft Silverlight, ASP.NET.

### 1.4. Краткие итоги

*Интернет* – глобальная телекоммуникационная сеть информационных и вычислительных ресурсов. В течение всего пяти лет *Интернет* достиг аудитории свыше 50-ти миллионов пользователей.

Различают логическую и физическую модели *Интернета*. Под логической прежде всего понимают Всемирную паутину ( *World Wide Web* ), а под физической – компьютеры, серверы и средства передачи данных между ними.

Чтобы различные компьютеры сети могли взаимодействовать, они должны использовать один и тот же протокол. Систему протоколов *Интернет* называют "стеком протоколов *TCP/IP* ".

*Всемирная паутина* – распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к *Интернету*.

*Всемирная паутина* стоит на "трех китах":

- язык гипертекстовой разметки документов *HTML* (HyperText Markup Language);
- универсальный способ адресации ресурсов в сети *URL* (Universal Resource Locator);



- протокол обмена гипертекстовой информацией *HTTP* (HyperText Transfer Protocol).

Концепции развития Всемирной паутины можно условно разделить на *Веб 1.0*, *Веб 2.0*, *Веб 3.0*.

*Веб-программирование* – это раздел программирования, ориентированный на разработку динамических *Интернет* -приложений.

Языки *веб-программирования* делятся на две группы: клиентские и серверные.

Клиентские языки обрабатываются на стороне пользователя (в основном в браузере).

Серверные языки программирования обрабатываются на стороне сервера.

Важной стороной работы серверных языков является Система управления базами данных (*СУБД*).

*HTML* – стандартный язык разметки документов во Всемирной паутин.

*XHTML* – язык разметки веб-страниц, по возможностям сопоставимый с *HTML*, созданный на базе *XML*.

*CSS* – технология описания внешнего вида документа, написанного языком разметки.

*XML* – текстовый формат, предназначенный для хранения структурированных данных, для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки.

*JavaScript* – это объектно-ориентированный скриптовый язык программирования.

*PHP* – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений.

*Perl* – высокоуровневый интерпретируемый динамический язык программирования общего назначения.

*AJAX* – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в "фоновом" обмене данными браузера с веб-сервером.

*Adobe Flash* и *Microsoft Silverlight* – мультимедийные платформы, используемые для создания *RIA-приложений*, а также для интеграции видеороликов в веб-страницы.

*ASP.NET* – технология создания веб-приложений и веб-сервисов от компании Microsoft.

## Литература

☐

1.

[Wikipedia. Интернет.](#)

☐

2.

Большой толковый словарь русского языка.

1998

☐

3.

[Глоссарий.ру. Словарь по естественным наукам](#)

☐

4.

[Энциклопедия «Кругосвет»](#)

☐

5.

RFC 1983

☐

6.

В.В. Лопатин

[Грамота.ру.](#)

Электронная версия «Русского орфографического словаря» Российской академии

☐

7.

[Студия «Артемия Лебедева»](#)

☐ 8.

[Служба русского языка](#)

☐ 9.

Mark Fischetti, Tim Berners-Lee

[Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web](#)

☐ 10.

Илья Ульянов И.

[IDC прогнозирует рост числа постоянных пользователей интернета](#)

☐ 11.

[История Интернета](#)

☐ 12.

Olexander Honcharuk

[Краткая история Интернета](#)

☐ 13.

Robert H'obbes' Zakon

[Hobbes' Internet Timeline 10](#)

☐ 14.

Денис Борн

[Интернету – 40 лет. Как все начиналось](#)

☐ 15.

Олифер В., Олифер Н.

[Введение в IP-сети](#)

☐ 16.

[Wikipedia. TCP/IP](#)

☐ 17.

Дуглас Камер

Сети TCP/IP, том 1. Принципы, протоколы и структура.

М.: «Вильямс»с. 88, 003

☐ 18.

[Wikipedia. Всемирная паутина](#)

☐ 19.

Лукач Ю.

[Справочник Веб-разработчика](#)

☐ 20.

Дэвид Оркард, Дэн Конноли, Крис Лилли, Марио Джекел, Ной Мендельсон, Норман Уолш, Пол Коттон, Рой Филдинг, Стюарт Уиллиамс., Тим Бернерс-Ли, Тим Брэй

[Architecture of the World Wide Web, Volume One](#)

☐ 21.

[Основные компоненты технологии World Wide Web](#)

☐ 22.

[Wikipedia. Веб 1.0](#)

☐ 23.

Dion Hinchcliffe

[All We Got Was Web 1.0, when Tim Berners-Lee Actually Gave Us Web 2.0](#)

☐ 24.

Jonathan Strickland.

[Is there a Web 1.0?](#)

☐ 25.

Тим О'Рейли

[Что такое Веб 2.0](#)

☐ 26.

[Wikipedia. Веб 2.0](#)

☐ 27.

[Wikipedia. Семантическая паутина](#)

☐ 28.

[Wikipedia. Веб 3.0](#)

☐ 29.

Джеймс Хендлер и Ора Лассила, Тим Бернерс-Ли

[Семантическая Сеть](#)

☐ 30.

[Консорциум W3](#)

☐ 31.

Джейсон Калаканис

[Web 3.0, «официальное» определение](#)

☐ 32.

[WEB 1.0 &gt; WEB 2.0 &gt; WEB 3.0 - куда движемся?](#)

☐ 33.

Marta Strickland

[The Evolution of Web 3.0](#)

☐ 34.

[Dal Web3.0 al Web4.0](#)

☐ 35.

[Wikipedia. Веб-программирование](#)

☐ 36.

[Wikipedia. HTML](#)

□ 37.

[Browser Market Share](#)

□ 38.

[Wikipedia. XHTML](#)

□ 39.

Билл Кароу, Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт

HTML, XHTML и CSS. Библия пользователя

3-е изд. – М.: «Диалектика», с. 762., 006

□ 40.

[Wikipedia. CSS](#)

□ 41.

Эрик А. Мейер

CSS-каскадные таблицы стилей: подробное руководство

М.: Символ, с. 57, 006

□ 42.

[Web Browser Standards Support Summary](#)

□ 43.

[CSS Compatibility and Internet Explorer](#)

□ 44.

[Wikipedia. XML](#)

□ 45.

Джефф Рафтер и др, Дэвид Хантер

XML. Базовый курс

М.: Вильямс, с. 1344, 009

□ 46.

David Megginson

[Imperfect XML: Rants, Raves, Tips, and Tricks ... from an Insider. Chapter 8](#)

☐ 47.

[Wikipedia. JavaScript](#)

☐ 48.

[Wikipedia. PHP](#)

☐ 49.

[TIOBE Software: Tiobe Index](#)

☐ 50.

[Wikipedia. Perl](#)

☐ 51.

[Comprehensive Perl Archive Network](#)

☐ 52.

[Wikipedia. AJAX](#)

☐ 53.

[Wikipedia. Adobe Flash](#)

☐ 54.

Mylene Mangalindan, Rebecca Buckman

[New Web-based Technology Draws Applications, Investors](#)

☐ 55.

[Wikipedia. Silverlight](#)

☐ 56.

[Wikipedia. ASP.NET](#)

☐ 57.

[Wikipedia. HTML](#)

□ 58.

Билл Кароу, Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт

HTML, XHTML и CSS. Библия пользователя

3-е изд. – М.: «Диалектика», с. 762., 2006

□ 59.

[Browser Market Share](#)

□ 60.

[HTML 3.2 Reference Specification](#)

□ 61.

[HTML 4.0 Specification](#)

□ 62.

[HTML 4.01 Specification](#)

□ 63.

[HTML5. Draft Standard – 23 February 2010](#)

□ 64.

Влад Мержевич

[HTML по шагам. Шаг 5. Теги HTML](#)

□ 65.

Влад Мержевич

[Структура документа](#)

□ 66.

[Wikipedia. DTD](#)

□ 67.

[Document Type Definition](#)

□ 68.



## Язык XML – практическое введение

☐ 69.

### Теги HTML. Тег META

☐ 70.

### Теги HTML. Тег LINK

☐ 71.

### Теги HTML. Тег SCRIPT

☐ 72.

### Теги HTML. Тег DIV

☐ 73.

Влад Мержевич

### Создание таблиц

☐ 74.

Влад Мержевич

### HTML по шагам. Шаг 6. Работа с текстом

☐ 75.

Влад Мержевич

### Как сделать ссылку

☐ 76.

Влад Мержевич

### Маркированные списки

☐ 77.

Влад Мержевич

### Вставка изображений

☐ 78.

[Wikipedia. Элементы HTML](#)

☐ 79.

[Wikipedia. XHTML](#)

☐ 80.

Влад Мержевич

[XHTML по шагам. Шаг 1. Что такое XHTML](#)

☐ 81.

Влад Мержевич.

[CSS по шагам](#)

☐ 82.

[Wikipedia. Доменное имя](#)

☐ 83.

[Wikipedia. ICANN](#)

☐ 84.

[Reserved Top Level DNS Names](#)

☐ 85.

[RU-CENTER. Как зарегистрировать домен](#)

☐ 86.

[R01](#)

☐ 87.

[Управление зоной домена](#)

☐ 88.

[Wikipedia. DNS](#)

☐ 89.

[Wikipedia. Хостинг](#)

□ 90.

[Wikipedia. Бесплатный хостинг](#)

□ 91.

[ФОСТАС. Об архитектуре программных и информационных систем](#)

□ 92.

[Software Engineering Institute. Software Architecture](#)

□ 93.

[IEEE Std. 610.12-1990](#)

□ 94.

[ГЛОССАРИЙ.ru](#)

□ 95.

[IEEE 1471](#)

□ 96.

Philippe Kruchten

The Rational Unified Process: An Introduction, Third Edition

Addison-Wesley Professional, 2005

□ 97.

Len Bass, Paul Clements, Rick Kazman

Software Architecture in Practice, Second Edition

Addison Wesley, 2003

□ 98.

[Object Management Group Inc. Unified Modeling Language](#)

□ 99.

et al, James McGovern

A Practical Guide to Enterprise Architecture

Prentice Hall, 2004

□ 100.

Питер Илес

[Что такое архитектура программного обеспечения](#)

□ 101.

[Архитектура вычислительного процесса](#)

□ 102.

Сергей Кузнецов

[Архитектуры ИС](#)

□ 103.

[Wikipedia. Файл-сервер](#)

□ 104.

Кузнецов С. Д.

[Общая классификация архитектур информационных приложений](#)

□ 105.

[Wikipedia. Клиент-сервер](#)

□ 106.

Андрей Крупин

[Архитектура информационных систем](#)

□ 107.

[Wikipedia. Трехуровневая архитектура](#)

□ 108.

Фаулер М.

Архитектура корпоративных программных приложений

2004

□ 109.

[Hot-ERP Project. Архитектуры информационных систем](#)

□ 110.

Алексей Федоров, Наталия Елманова

[Архитектура современных Web-приложений](#)

□ 111.

[Wikipedia. Сервис-ориентированная архитектура](#)

□ 112.

[Wikipedia. Шаблоны проектирования](#)

□ 113.

Alexander K. et al.

Pattern Language

Oxford, 977

□ 114.

Дж. Влиссидес, Р. Джонсон, Р. Хелм, Э. Гамма

Приемы объектно-ориентированного проектирования. Паттерны проектирования.

СПб: «Питер», с. 366., 007

□ 115.

Ольга Дубина

[Обзор паттернов проектирования](#)

□ 116.

[GET и POST: что лучше](#)

□ 117.

[КОМТЕТ. Различие между методами GET и POST](#)

□ 118.

[Wikipedia. HTTP](#)

☐ 119.

[SEO-студия «Оптимизация веб-сайтов». GET или POST? Сравнительный анализ методов передачи данных](#)

☐ 120.

[Блог простого программиста. POST или GET?](#)

☐ 121.

Столбовский Д.Н.

[Разработка Web-приложений ASP .NET с использованием Visual Studio .NET](#)

☐ 122.

[Wikipedia. .NET](#)

☐ 123.

Дмитрий Андреев

[Архитектурный обзор прикладной платформы .NET](#)

☐ 124.

[MSDN. Общие сведения об основных понятиях платформы .NET Framework](#)

☐ 125.

[WebRegion. Основы Windows Presentation Foundation \(WPF\)](#)

☐ 126.

[Wikipedia. Windows Presentation Foundation](#)

☐ 127.

[Wikipedia. Windows Communication Foundation](#)

☐ 128.

[Wikipedia. Windows Workflow Foundation](#)

☐ 129.

[Wikipedia. Windows CardSpace](#)

□ 130.

[Visual Studio 2008 and .NET 3.5 Released](#)

□ 131.

[MSDN. Общие сведения о ASP.NET](#)

□ 132.

[MSDN. Жизненный цикл ASP.NET](#)

□ 133.

Peter Bromberg

[ASP.NET 2.0 Page LifeCycle](#)

□ 134.

[MSDN. Общие сведения об управлении состоянием ASP.NET](#)

□ 135.

Марио Шпушта, Мэтью Мак-Дональд

[Управление состоянием](#)

□ 136.

[MSDN. Общие сведения о веб-страницах ASP.NET](#)

□ 137.

[MSDN. Стандартные элементы управления панели элементов](#)

□ 138.

Троелсен Э.

C# и платформа .NET. Библиотека программиста.

СПб.: Питер, 796 с.: ил., 2004

□ 139.

Столбовский Д.Н.

[Разработка Web-приложений ASP .NET с использованием Visual Studio .NET](#)

□ 140.

[MSDN. ASP.NET Master Pages Overview](#)

□ 141.

[MSDN. Серверные элементы управления ASP.NET](#)

□ 142.

[MSDN. Навигационные элементы управления панели элементов](#)

□ 143.

[MSDN. Общие сведения об элементе управления Menu](#)

□ 144.

[MSDN. Общие сведения о серверном веб-элементе управления SiteMapPath](#)

□ 145.

[MSDN. Общие сведения о серверном веб-элементе управления GridView](#)

□ 146.

[MSDN. Создание пользовательского столбца в серверном веб-элементе управления GridView](#)

□ 147.

[MSDN. События серверного веб-элемента управления GridView](#)

□ 148.

[MSDN. Общие сведения о серверном веб-элементе управления ListView](#)

□ 149.

[MSDN. Пользовательские элементы управления ASP.NET](#)

□ 150.

[Wikipedia. AJAX](#)

□ 151.

[MSDN. Добавление клиентских возможностей и возможностей AJAX](#)

□ 152.



[MSDN. Общие сведения о ASP.NET AJAX](#)

☐ 153.

[ASP.NET MVC на реальном примере. Теория и вступление](#)

☐ 154.

Гайдар Магдануров

[Введение в MVC Framework](#)

☐ 155.

[Wikipedia. ASP.NET MVC Framework](#)

☐ 156.

Крис Таварес

[Создание веб-приложений без форм](#)

☐ 157.

Савельева Н.В.

[Базы данных и СУБД. Введение в SQL](#)

☐ 158.

[CodeNet. Введение в базы данных](#)

☐ 159.

[Ст. 1259 IV часть Гражданского Кодекса России](#)

☐ 160.

Когаловский М.Р.

Энциклопедия технологий баз данных.

М.: Финансы и статистика, 002

☐ 161.

[ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными](#)

☐ 162.

Дейт К. Дж.

Введение в системы баз данных

8-е издание.: Пер. с англ. – М.: Вильямс, 2005

□ 163.

IEEE Standard Glossary of Software Engineering Terminology IEEE Std. 610.12-1990

□ 164.

Бегг К., Коннолли Т.

Базы данных. Проектирование, реализация и сопровождение. Теория и практика

2003

□ 165.

[ГЛОССАРИЙ.ru](http://glossary.ru)

□ 166.

[Wikipedia. База данных](#)

□ 167.

[Wikipedia. Система управления базами данных](#)

□ 168.

Зеленков Ю.А.

[Введение в базы данных](#)

□ 169.

Кренке Д.

[Теория и практика построения баз данных](#)

□ 170.

[Wikipedia. Модель данных](#)

□ 171.

Codd E.F.

Data Models in Database Management. Proc. Workshop in Data Abstraction, Databases, and Conceptual Modelling , ACM SIGPLAN Notices 16(1)

1981

□ 172.

[Библиотека программиста. Основы реляционных Баз данных](#)

□ 173.

[Wikipedia. Реляционная модель данных](#)

□ 174.

[Wikipedia. Объектно-ориентированная база данных](#)

□ 175.

[Wikipedia. Нормальная форма](#)

□ 176.

Пушников А.Ю.

[Транзакции и целостность баз данных](#)

□ 177.

[Wikipedia. ACID](#)

□ 178.

[Wikipedia. Проектирование баз данных](#)

□ 179.

Когаловский М.Р.

Энциклопедия технологий баз данных.

М.: Финансы и статистика, 800 с. 2002

□ 180.

Бегг К., Коннолли Т.

Базы данных. Проектирование, реализация и сопровождение. Теория и практика

3-е изд. – М.: Вильямс, 1436 с. 003

□ 181.

[Библиотека программиста. Основы использования WWW – технологий для доступа к существующим базам данных](#)

□ 182.

[Библиотека программиста. ЯЗЫК SQL](#)

□ 183.

[Wikipedia. SQL](#)

□ 184.

Алексей Федоров, Наталия Елманова.

[Введение в язык SQL](#)

□ 185.

O'Reilly

[Network An Interview with Chris Date](#)

□ 186.

[MSDN. Обзор SQL Server](#)

□ 187.

Мишель Дамлер

[Microsoft SQL Server 2008](#)

□ 188.

Майкл Оти

[Обзор SQL Server 2008](#)

□ 189.

[MSDN. Общие сведения об интеграции со средой CLR](#)

□ 190.

Столбовский Д.Н.

[Основные сведения о модели доступа к данным ADO.NET](#)

□ 191.

Троелсен Э.

C# и платформа .NET. Библиотека программиста

СПб.: Питер, 796 с.: ил.2004

□ 192.

[Документация по серверу Microsoft SQL Server. Доступ к данным](#)

□ 193.

[MSDN. Доступ к данным с помощью ASP.NET](#)

□ 194.

[Wikipedia. Language Integrated Query](#)

□ 195.

[MSDN. LINQ](#)

□ 196.

[Блог одного человека. LINQ](#)

□ 197.

Троелсен Э

C# и платформа .NET. Библиотека программиста

СПб.: Питер, 796 с.: ил.2004

□ 198.

[MSDN. Общее руководство программирования на LINQ](#)

□ 199.

Влад Чистяков

[LINQ как шаг к функциональному программированию](#)

□ 200.

Иван Бодяжин

[C# 3.0 и LINQ](#)

□ 201.

[MSDN. LINQ to SQL](#)

□ 202.

[Wikipedia. ADO.NET Entity Framework](#)

□ 203.

[MSDN. ADO.NET Entity Framework](#)

□ 204.

Роман Здебский

[Сравнительный анализ LINQ, ADO.NET Entity Framework и ADO.NET Data Services](#)

□ 205.

Джон Папа

[Обзор ADO.NET Entity Framework](#)

□ 206.

[Domain-Driven Design Community](#)

□ 207.

Дэвид Лэриби

[Введение в проблемно-ориентированное проектирование](#)

□ 208.

[Wikipedia. Domain-driven design](#)

□ 209.

Е., Evans

Domain-Driven Design – Tackling Complexity in the Heart of Software

2004, Addison-Wesley

□ 210.

Андрей Бибичев

[Domain-Driven Design for Java Developers](#)

□ 211.

Artur Trosin

[Domain Driven Design: Learning](#)

□ 212.

[.NET Domain-Driven Design with C#: Problem-Design-Solution](#)

□ 213.

Billy McCafferty

[Sharp Architecture: ASP.NET MVC Best Practices with NHibernate](#)

□ 214.

Artur Trosin

[C# Domain-Driven Design sample application](#)

□ 215.

Джимми Нильссон.

Применение DDD и шаблонов проектирования. Проблемно-ориентированное проектирование приложений с примерами на C# и .NET.

М.: Вильямс, 560 стр.2008

□ 216.

Дэвид Флэнаган

JavaScript. Подробное руководство

СПб.: Символ-Плюс, с. 992.2008

□ 217.

[Wikipedia. JavaScript](#)

☐ 218.

Douglas Crockford

[The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language](#)

☐ 219.

[Повторное введение в JavaScript](#)

☐ 220.

[Wikipedia. JSON](#)

☐ 221.

[Wikipedia. AJAX](#)

☐ 222.

[Wikipedia. Comet](#)

☐ 223.

[Проект eyeOS](#)

☐ 224.

[Wikipedia. Библиотека JavaScript](#)

☐ 225.

[ExtJS](#)

☐ 226.

[jQuery](#)

☐ 227.

[Dojo Toolkit](#)

☐ 228.

[Wikipedia. JSON](#)

☐ 229.



## [ExtJS](#)

□ 230.

## [Wikipedia. ExtJS](#)

□ 231.

## [Объектно-ориентированный дизайн ExtJS](#)

□ 232.

Colin Ramsay, Shea Frederick, Steve 'Cutter' Blades

Learning Ext JS

2008

□ 233.

Александр Лозовюк

## [Общий взгляд на AJAX фреймворк ExtJS](#)

□ 234.

## [Wikipedia. Prototype](#)

□ 235.

Дмитрий Николаев

## [Введение в Prototype](#)

□ 236.

## [Wikipedia. jQuery](#)

□ 237.

## [jQuery](#)

□ 238.

## [Visual jQuery](#)

□ 239.

Dmitrii Dimandt

[jQuery – Javascript нового поколения](#)

□ 240.

[Wikipedia. Comet \(программирование\)](#)

□ 241.

Максим Ищенко

[Асинхронный HTTP, но не AJAX](#)

□ 242.

[Sontan. Интернет в реальном времени и Comet](#)

□ 243.

James Simpson

[Scalable COMET Combined with ASP.NET](#)

□ 244.

Троелсен Э.

Язык программирования C# 2008 и платформа .NET 3.5

М.: Вильямс, с. 1344.2010

□ 245.

[MSDN. Веб-службы в управляемом коде](#)

□ 246.

[MSDN. Использование Веб-служб в технологии AJAX ASP.NET](#)

□ 247.

[MSDN. Добавление клиентских возможностей и возможностей AJAX](#)

□ 248.

Олег Шаститко

[Использование AJAX в ASP.NET](#)

□ 249.

Нуньес Тельес Кирилл

[Методы продвижения сайта](#)

☐ 250.

Налютин Н.Ю., Сеницын С.В.

[Семинар: Тестирование в Microsoft Solutions Framework](#)

☐ 251.

[MSDN. Вызов веб-служб из клиентского сценария](#)

☐ 252.

[MSDN. Создание Веб-служб в управляемом коде](#)

☐ 253.

Александр Левашов

[Опыт применения RIA в бизнесе](#)

☐ 254.

Константин Ковалев

[RIA – значит свобода](#)

☐ 255.

[DATAART. Обогащенные Интернет приложения](#)

Upper Saddle River: Prentice Hall, с. 49-94.2001

☐ 256.

Илья Воробьев

[Выбирая между WPF и Silverlight](#)

Upper Saddle River: Prentice Hall, с. 49-94.2001

☐ 257.

[Microsoft Silverlight 2 релиз](#)

☐ 258.

[MSDN. About Native XMLHTTP](#)

□ 259.

[Jasp Project](#)

□ 260.

[Wikipedia. RIA](#)

□ 261.

Simon Whatley

[Rich Internet Applications – A Background](#)

□ 262.

[Adobe Flash Platform ActionScript reference for RIA development](#)

□ 263.

[Wikipedia. Adobe Flash](#)

□ 264.

[Wikipedia. Adobe Flex](#)

□ 265.

Константин Ковалев

[Сравнение современных технологий для создания насыщенных интернет-приложений \(RIA\)](#)

Harvard Business Press, с. 25.2001

□ 266.

[Adobe Flex Builder 3](#)

Addison Wesley, с. 61.2003

□ 267.

[Wikipedia. Adobe Integrated Runtime](#)

□ 268.

[Технология ActiveX](#)

□ 269.

[Wikipedia. Windows Presentation Foundation](#)

□ 270.

[Windows Presentation Foundation](#)

□ 271.

Константин Шишкин

[Основы Windows Presentation Foundation \(WPF\)](#)

□ 272.

Мэтью Мак-Дональд

[WPF: Windows Presentation Foundation в .NET 3.5 с примерами на C# 2008 для профессионалов](#)

2-ое. – М.: «Вильямс», с. 928.2008

□ 273.

[MSDN. Знакомство с Windows Presentation Foundation](#)

□ 274.

[Wikipedia. Silverlight](#)

□ 275.

[Jasp Project Hosting on Google Code](#)

□ 276.

Михаил Черномордиков

[Silverlight 3 Beta – все ссылки и возможности](#)

□ 277.

[Европейский институт стандартов по электросвязи. Словарь терминов по безопасности и криптографии](#)

□ 278.

[Wikipedia. JavaFX](#)

□ 279.

Леонид Черняк

[JavaFX – технология богатых приложений](#)

□ 280.

[JavaFX](#)

□ 281.

Александр Шер

[Еще одно сравнение Adobe Flash и Silverlight 2](#)

□ 282.

Dan Farber

[JavaOne: Sun rolls out JavaFX](#)

□ 283.

Florian Moritz

[Rich Internet Applications \(RIA\): A Convergence of User Interface Paradigms of Web and Desktop Exemplified by JavaFX](#)

□ 284.

Михаил Черномордилов

[Луч серебряного света](#)

□ 285.

[ГОСТ Р 50922-2006. Национальный стандарт РФ «Защита информации. Основные термины и определения»](#)

□ 286.

[ГОСТ Р ИСО/МЭК 17799-2005. Национальный стандарт РФ «Информационная технология. Практические правила управления информационной безопасностью»](#)

□ 287.

[ГЛОССАРИЙ.ru](#)

□ 288.

[Wikipedia. Информационная безопасность](#)

□ 289.

[Web Application Security Consortium. Security Statistics Projects](#)

М.: Диалектика, с. 272. 2007

□ 290.

[MSDN. Microsoft Silverlight 3. Описание продукта](#)

М.: Диалектика, с. 272. 2007

□ 291.

[ГОСТ Р ИСО/МЭК 13335-1-2006. Национальный стандарт РФ «Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий»](#)

□ 292.

Домарев В. В.

[Безопасность информационных технологий. Системный подход](#)

К.: ООО ТИД Диа Софт, 992 с.2004

□ 293.

Лапин В. И., Лапина М. А., Ревин А. Г.

[Информационное право.](#)

М.: ЮНИТИ-ДАНА, Закон и право, 2004

□ 294.

Михаил Кадер

[Обзор сетевых атак](#)

□ 295.

[Wikipedia. Хакерская атака](#)

□ 296.

[The Web Application Security Consortium. Threat Classification](#)

□ 297.

[Neko-art-line. Возможные виды атак на Web-приложения и возможные способы их предотвращения](#)

□ 298.

[Wikipedia. Межсайтовый скриптинг](#)

□ 299.

[Security Lab. Отчет по уязвимостям за первый квартал 2008 года](#)

□ 300.

[Р 50.1.053-2005. Рекомендации по стандартизации «Информационные технологии. Основные термины и определения в области технической защиты информации»](#)

□ 301.

Джеймс Хендлер и Ора Лассила, Тим Бернерс-Ли

[The Semantic Web. Scientific American](#)

□ 302.

Дэвид Росс

[Безопасность IE8: XSS-фильтр](#)

□ 303.

David Ross

[IE8 Security Part IV: The XSS Filter](#)

□ 304.

Амит Клейн

[Третий тип XSS: Межсайтовый скриптинг через DOM](#)

□ 305.

Дмитрий Евтеев, Сергей Гордейчик

[Защита Internet Explorer 8. Анализ эффективности](#)



М.: Вильямс, с. 1408.2009

□ 306.

[Wikipedia. Фишинг](#)

М.: Вильямс, с. 1408.2009

□ 307.

[IE8 и SmartScreen фильтр](#)

□ 308.

Эрик Лоуренс

[Безопасность IE8: фильтр SmartScreen](#)

□ 309.

Алекс Гловер

[Изменения в фильтре SmartScreen в IE8 RC1](#)

□ 310.

Alex Glover

[IE8 Security Part VIII: SmartScreen Filter Release Candidate Update](#)

□ 311.

Eric Lawrence

[IE8 Security Part III: SmartScreen Filter](#)

□ 312.

Эрик Лоуренс

[Настоящая защита с фильтром SmartScreen](#)

□ 313.

Eric Lawrence

[Real-World Protection With IE8's SmartScreen Filter](#)

□ 314.

[Wikipedia. Data Execution Prevention](#)

□ 315.

[MSDN. Data Execution Prevention](#)

□ 316.

Эрик Лоуренс

[Безопасность IE8: технология защиты памяти DEP \(NX\)](#)

□ 317.

Eric Lawrence

[IE8 Security Part I: DEP/NX Memory Protection](#)

□ 318.

Денис Урасов

[Безопасность IE8: технология защиты памяти DEP \(NX\)](#)

□ 319.

[Wikipedia. SSL](#)

□ 320.

Семенов Ю.А.

[Процедуры, диагностики и безопасность в Интернет](#)

□ 321.

[Wikipedia. HTTPS](#)

□ 322.

Диан Стотлемайер

[Тестирование Web-приложений.](#)

М.: «КУДИЦ-Образ», с. 240.2003

□ 323.

Diane Stottlemeyer

[Automated Web Testing Toolkit. Expert Methods for Testing and Managing Web Applications](#)

John Wiley & Sons, 2001

□ 324.

Налютин Н.Ю., Сеницын С.В.

[Верификация программного обеспечения](#)

Русская Редакция, 004

□ 325.

Илья Бромберг.

[Автоматизация тестирования](#)

М.: Вильямс, с. 1344. 2010

□ 326.

[Форум портала для IT-менеджеров: \[www.it4business.ru\]\(http://www.it4business.ru\). Автоматизированное тестирование ПО](#)

М.: Вильямс, с. 1344. 2010

□ 327.

Сортов А.А., Хорошилов А.В.

[Функциональное тестирование Web-приложений на основе технологии UniTesK](#)

□ 328.

[ProLAN. Управление бизнес приложениями клиента, работающими в сети NSP](#)

□ 329.

Алексей Булат, Владимир Антонов, Ирина Тетерук

[Тестирование удобства пользования \(Usability Testing\)](#)

□ 330.

Nielsen J.

[Ten Usability Heuristics](#)

□ 331.

[IT-Online. Общий оценочный лист тестирования usability web-сайта](#)

□ 332.

Александр Лозовюк

[Обзор решений для тестирования сайтов](#)

□ 333.

Александр Лозовюк

[Решения для тестирования веб-сайтов](#)

Prentice Hall, с. 22-24.2008

□ 334.

[Тестирование web-приложений](#)

17th Annual International Conference of the Association of Management

□ 335.

[Positive Technologies. Статистика уязвимости Web-приложений за 2007 год](#)

McGraw-Hill2, 000

□ 336.

Андрей Широбоков

[Нагрузочное тестирование \(Виды тестирования производительности\)](#)

McGraw-Hill2, 000

□ 337.

[Нагрузочное тестирование](#)

□ 338.

Сергей Рогов

[Тестирование производительности Web-серверов](#)

□ 339.

Комал Мирчандани, Шрути Уджвал

[Использование. IBM Rational Functional Tester для автоматизированного тестирования глобализованных приложений](#)

□ 340.

[MSDN. Знакомство со средствами разработчика в Internet Explorer 8](#)

□ 341.

Налютин Н.Ю., Сеницын С.В.

[Тестирование пользовательского интерфейса](#)

□ 342.

Петр Можаяев

[Средства автоматизированного тестирования](#)

□ 343.

[Wikipedia. HP LoadRunner](#)

□ 344.

Александр Лозовюк

[Обзор решений для тестирования сайтов](#)

□ 345.

[Firebug](#)

□ 346.

[Средства разработки в IE8](#)

М.: Вильямс, . – с. 1344. 010

□ 347.

[Wikipedia. Отладка](#)

М.: Вильямс, с. 272.2009

□ 348.

[Wikipedia. Javascript](#)

СПб.: Символ-Плюс, с. 752.2008

□ 349.

[Wikipedia. Профилирование](#)

Harvard Business School Press, с. 21.2001

□ 350.

[CNews. IE8: как облегчить труд веб-разработчика](#)

Harvard Business School Press, с. 21.2001

□ 351.

[ProLAN. Управление производительностью приложений на принципах SLA](#)

□ 352.

Martin Mihaylov

[The MultiscaleImage control and the SubImages collection](#)

□ 353.

Cory Doctorow

[Metacrap: Putting the torch to seven straw-men of the meta-utopia](#)

□ 354.

Новиков С.В.

[Semantic Web, поиск на основе бинарных моделей, проблемы современных поисковых агентов](#)

□ 355.

Tim Berners-Lee

[Linked Data](#)

□ 356.

Леонид Черняк

[О стеке стандартов Semantic Web](#)

□ 357.

[Wikipedia. Semantic Web](#)

□ 358.

Андреев А.М. Березкин Д.В. Рымарь В.С. Симаков К.В.

[Использование технологии Semantic Web в системе поиска несоответствий в текстах документов](#)

□ 359.

[Русскоязычное сообщество веба данных](#)

□ 360.

Нэвин Бэлэни

[Будущее Web – за семантикой](#)

□ 361.

[OWL Web Ontology Language Reference](#)

□ 362.

[Dublin Core Metadata Initiative \(DCMI\)](#)

□ 363.

Михаил Навернюк

[Перспективы формирования Семантической Сети](#)

□ 364.

Вячеслав Тихонов

[Поисковые системы в сети Интернет](#)

□ 365.

Колисниченко Д. Н.

[Поисковые системы и продвижение сайтов в Интернете.](#)

М.: Диалектика, с. 272. 2007

□ 366.

[MSDN. Спецификация формата веб-фрагмента – версия 0.9](#)

□ 367.

[Tim Berners-Lee. The Semantic Web as a language of logic](#)

□ 368.

James Hendler, Ora Lassila

[Embracing «Web 3.0»](#)

□ 369.

[Wikipedia. Микроформаты](#)

□ 370.

John Allsopp

[Microformats: Empowering Your Markup for Web 2.0](#)

□ 371.

[Microformats wiki](#)

□ 372.

[Введение в микроформаты](#)

М.: ЮНИТИ-ДАНА, Закон и право, 2004

□ 373.

[CNews. Microsoft вновь изобретает Web 3.0](#)

К.: ООО ТИД Диа Софт, 992 с. 2004

□ 374.

Дарон Йондем

[Новые функции для веб-фрагментов, хранения и повышения  
производительности веб-приложений](#)

К.: ООО ТИД Диа Софт, 992 с. 2004

□ 375.

[Wikipedia. Поисковая система](#)



□ 376.

[Wikipedia. Семантические веб-сервисы](#)

М.: Диалектика, с. 272. 2007

□ 377.

Андрей Крупин

[Инструменты для мониторинга посещаемости сайтов](#)

□ 378.

[DMG-Studio. Веб-аналитика. Google Analytics или Яндекс. Метрика](#)

□ 379.

[Все поисковые системы \(поисковики\)](#)

□ 380.

Козинев Е.А., Мееров И.Б., Сысоев А.В.

[Технологии программирования на базе Microsoft Solutions Framework](#)

□ 381.

[Wikipedia. Microsoft Solutions Framework](#)

□ 382.

Онвард

[Методы продвижения сайта](#)

□ 383.

[Wikipedia. Веб-аналитика](#)

□ 384.

Applegate L. M.

[E-Business Models: Making Sense of the Internet Business Landscape. In W. G. Dickson & G. DeSanctis \(Eds.\), Information technology and the future enterprise: New models for managers.](#)

Upper Saddle River: Prentice Hall, с. 49-94.2001

□ 385.

[Рейтинг поисковых систем 2009](#)

Upper Saddle River: Prentice Hall, с. 49-94.2001

□ 386.

[Wikipedia. OpenSearch](#)

□ 387.

[OpenSearch](#)

□ 388.

[Microsoft Solutions Framework. Модель процессов MSF вер. 3.1](#)

□ 389.

[Проект «ITLIFT.RU». Бизнес-модель](#)

□ 390.

[Microsoft Start in Garage. Курс технологичного предпринимательства](#)

□ 391.

Дмитрий Ландэ

[Поисковые системы: поле боя – семантика](#)

□ 392.

[CEO. Как раскрутить свой сайт самому](#)

□ 393.

[MSF for Agile Software Development Process Guidance](#)

2-ое. – М.: «Вильямс», с. 928.2008

□ 394.

[Microsoft Solutions Framework. Модель проектной группы MSF вер. 3.1](#)

2-ое. – М.: «Вильямс», с. 928.2008

□ 395.

[Wikipedia. Веб-разработка](#)

□ 396.

Михаэль Раппа

[Бизнес модели в Интернете](#)

□ 397.

Rappa M.

[Business Models on the Web](#)

□ 398.

P., Timmers

[Business Models for Electronic Markets](#)

□ 399.

and C. G. Traver, Laudon K. C.

[E-Commerce: Business, Technology, Society](#)

Addison Wesley, c. 61.2003

□ 400.

and M. Vitale, Weill P.

[Place to Space: Migrating to E-Business Models](#)

Harvard Business Press, c. 25.2001

□ 401.

[Wikipedia. Online advertising](#)

Harvard Business Press, c. 25.2001

□ 402.

[MSDN. Deep Zoom](#)

□ 403.

David Ross

[IE 8 XSS Filter Architecture / Implementation](#)

□ 404.

[Wikipedia. HTML](#)

□ 405.

Евгений Скрипец

[Каскадные Таблицы Стилей](#)

□ 406.

and J. Kador., Hartman A., J. Sifonis

[Net Ready: Strategies for Success in the E-Conomy](#)

McGraw-Hill2, 000

□ 407.

Viehland D.

[New Business Models for Electronic Commerce](#)

17th Annual International Conference of the Association of Management

□ 408.

and D. Viehland, D. King, E., J. Lee, J. McKay, P. Marshall, Turban

[Electronic Commerce 2008: A Managerial Perspective](#)

Prentice Hall, c. 22-24.2008

□ 409.

[Tina Clarke. Dynamic Web Templates](#)

□ 410.

[Microsoft. Create a Dynamic Web Template](#)

□ 411.

[MSDN. Dynamic web templates overview](#)

□ 412.

[Dynamic Web Template Interchange Guidelines](#)

□ 413.

[Microsoft.Создание эталонного макета веб-страницы с помощью динамических веб-шаблонов](#)

□ 414.

[MSDN. Отладка сценариев HTML и CSS помощью средств разработчика](#)

□ 415.

[MSDN. LINQ to SQL](#)

□ 416.

[MSDN. LINQ](#)

□ 417.

[Консорциум W3](#)

□ 418.

Vitale M., Weill P.

[Place to Space: Migrating to E-Business Models](#)

Harvard Business School Press, с. 21.2001

□ 419.

Билл Кеннеди, Чак Муссиано

[HTML и XHTML. Подробное руководство](#)

СПб.: Символ-Плюс, с. 752.2008

□ 420.

Камерон Молл, Саймон Коллизон, Энди Бадд

[CSS. Профессиональное применение Web-стандартов](#)

М.: Вильямс, с. 272.2009

□ 421.

Троелсен Э.

[Язык программирования C# 2008 и платформа .NET 3.5](#)

М.: Вильямс, . – с. 1344. 010

☐ 422.

[MSDN. Общие сведения о жизненном цикле веб-страниц ASP.NET](#)

☐ 423.

[MSDN. Общие сведения о ASP.NET](#)

☐ 424.

[MSDN. Создание веб-страниц ASP.NET](#)

☐ 425.

[MSDN. Общие сведения о жизненном цикле веб-страниц ASP.NET](#)

☐ 426.

[MSDN. SQL Server и ADO.NET](#)

☐ 427.

[Microsoft.Изменение разметки страницы-шаблона с помощью динамических веб-шаблонов](#)

☐ 428.

[Wikipedia. CSS](#)

☐ 429.

[OpenSearch](#)

М.: Вильямс, с. 1344. 2010

☐ 430.

[Wikipedia. OpenSearch](#)

☐ 431.

Jaime Rodriguez

[Working with Collections in Deep Zoom](#)

☐ 432.

[ExtJS 3.1.1 – API Documentation](#)

□ 433.

[MSDN. Добавление клиентских возможностей и возможностей AJAX](#)

□ 434.

[MSDN. Модель кода для доступа к веб-службам в управляемом коде](#)

□ 435.

[MSDN. Общие сведения о частичной отрисовке страниц](#)

□ 436.

Марио Шпушта, Мэтью Мак-Дональд

[Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов](#)

М.: Вильямс, с. 1408.2009

□ 437.

Александр Порубов

[DeepZoom в Silverlight 2](#)

□ 438.

[Microsoft. Deep Zoom Composer](#)

□ 439.

[MSDN. Подписка на содержимое с помощью веб-фрагментов](#)

□ 440.

Михаил Черномордилов

[Разработка простого WebSlice для IE8](#)

□ 441.

[Wikipedia. Микроформаты](#)

□ 442.

Троелсен Э.

[Язык программирования C# 2008 и платформа .NET 3.5](#)

М.: Вильямс, с. 1344. 2010

☐ 443.

[Разработка Web-сервисов XML и серверных компонентов на Microsoft Visual Basic .NET и Microsoft Visual C#.NET. Учебный курс MCAD / MCDS](#)

Русская Редакция, 004

☐ 444.

[MSDN. Вызов веб-служб из клиентского сценария](#)

John Wiley & Sons, 2001

☐ 445.

[Создаем визуальный поиск в IE8 для поиска друзей в Twitter](#)

М.: «КУДИЦ-Образ», с. 240.2003

☐ 446.

[Блог команды разработчиков Internet Explorer](#)

М.: «КУДИЦ-Образ», с. 240.2003

☐ 447.

Себастьян Циммерман

[Hello, World или начинаем работать с IE8 Visual Search](#)

☐ 448.

[MSDN. Расширение поставщиков поиска в Internet Explorer](#)

☐ 449.

[MSDN. Обзоры веб-фрагментов](#)

☐ 450.

[MSDN. Ускорители OpenService: руководство разработчика](#)

☐ 451.

[Microformats wiki](#)



□ 452.

Константин Кичинский

[Deep Zoom. Создание интерактивной коллекции фотографий за 15 минут](#)