

Лекция №1 Основы World Wide Web (WWW)

1. Основные понятия и термины.
2. Стандарты WWW
3. Принципы организации DNS и URL
4. Описание протокола HTTP
- 5 Стандарт CGI
- 6 Технологии WWW
- 7 Программное обеспечение WWW
- 8 Браузеры
- 9 ТОП-10 поисковых систем мира
- 10 Средства редактирования HTML-кода

1. Основные понятия и термины

Интернет - это глобальная компьютерная сеть, объединяющая локальные сети предприятий и организаций и компьютеры отдельных пользователей в общее информационное пространство. Интернет использует для связи магистральные каналы поставщиков телекоммуникационных услуг и клиентские каналы.

Интернет представляет свою инфраструктуру для прикладных сервисов различного назначения (файлообменных, почтовых, гипертекстовых, VoIP и проч.)

World Wide Web (WWW, web, веб, Всемирная паутина) — это распределенная информационная система, предоставляющая доступ к гипертекстовым документам по протоколу **HTTP (HyperText Transfer Protocol)**. **WWW** - это сетевая технология, построенная на клиент-серверной архитектуре и использующая инфраструктуру Интернет для взаимодействия между сервером и клиентом (рис. 1).

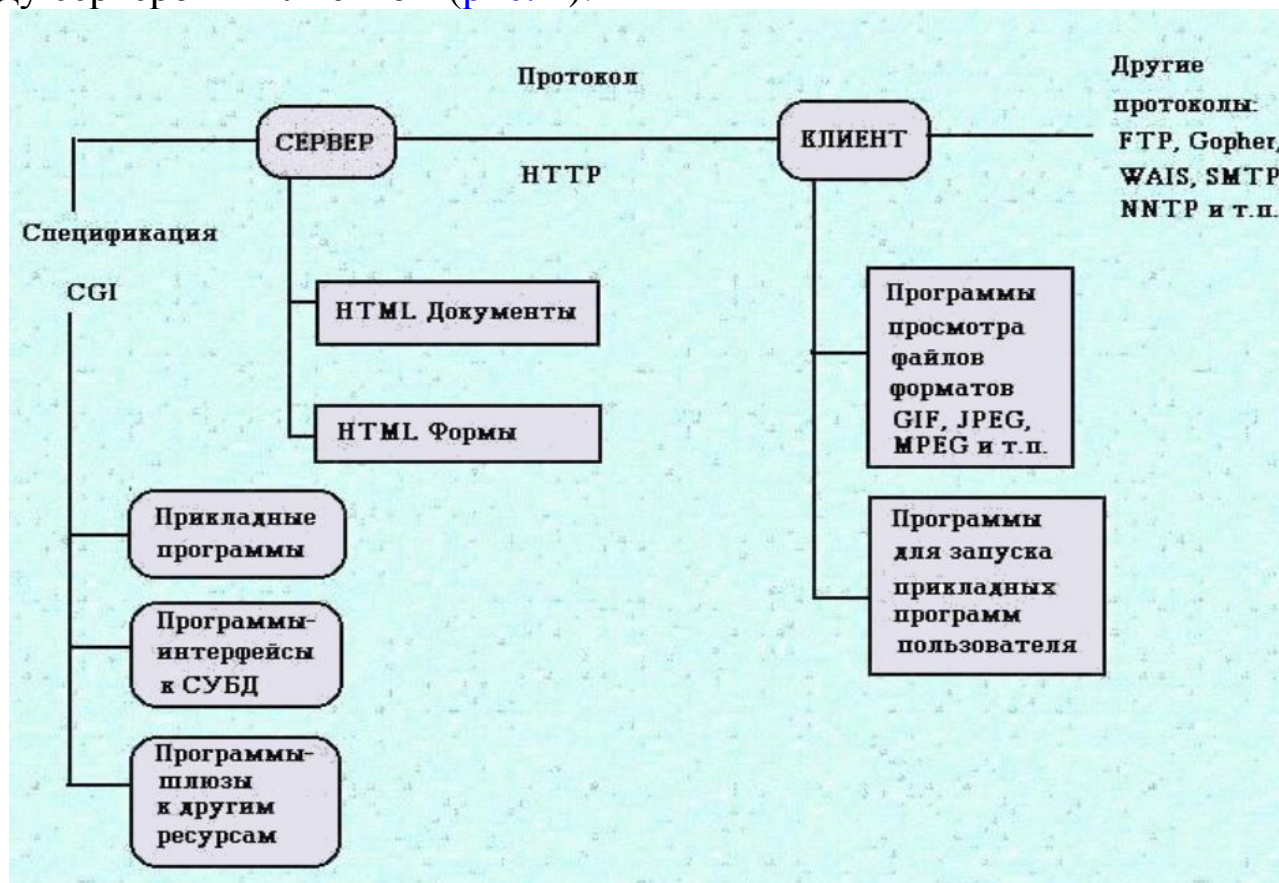


Рис.1. Клиент-серверная архитектура WWW

Тим Бернерс-Ли, Сэр Тимоти Джон Бернерс-Ли — британский учёный-физик, изобретатель концепции Всемирной паутины (совместно с Робертом Кайо), автор URL, HTTP и HTML, действующий глава Консорциума Всемирной паутины (W3C), автор концепции семантической паутины и множества других разработок в области информационных технологий. Проект был предложен в 1989 году.

Web-страницы- документы, представленные в виде гипертекста.

HTML (HyperText Markup Language — «язык разметки гипертекста») — стандартный язык разметки документов во Всемирной паутине.

Большинство веб-страниц созданы при помощи языка HTML. Язык HTML интерпретируется браузером и отображается в виде документа в удобной для человека форме.

Несколько Web-страниц, объединенных общей тематикой, оформлением, связанных гипертекстовыми ссылками и обычно находящихся на одном и том же Web-сервере, называются **Web-сайтом**.

В 1993 появился первый браузер.

Браузер - это программа, способная понимать гипертекст и отображать содержимое веб-страниц. Предназначен для загрузки и просмотра информации с Web-сайтов. Браузер (Internet Explorer, Opera ...) является мультипротокольным клиентом и интерпретатором HTML. И как типичный интерпретатор, клиент в зависимости от команд (тегов) выполняет различные функции. В круг этих функций входит не только размещение текста на экране, но обмен информацией с сервером по мере анализа полученного HTML-текста, что наиболее наглядно происходит при отображении встроенных в текст графических образов.

Серверы WWW (Web-серверы) - это хранилища гипертекстовой (в общем случае) информации.

Сервер HTTP (Apache, IIS ...) обрабатывает запросы клиента на получение файла (в самом простом случае).

В основе WWW - взаимодействие веб-сервера и браузеров по специальному протоколу (HTTP).

Браузер может обратиться к веб-серверу по доменному имени или по IP-адресу, передавая в запросе идентификатор требуемого ресурса. Получив запрос от клиента, сервер находит соответствующий ресурс на локальном устройстве хранения и отправляет его как ответ. Браузер принимает ответ и обрабатывает его соответствующим образом, в зависимости от типа ресурса (отображает гипертекст, показывает изображения, сохраняет полученные файлы и т.п.).

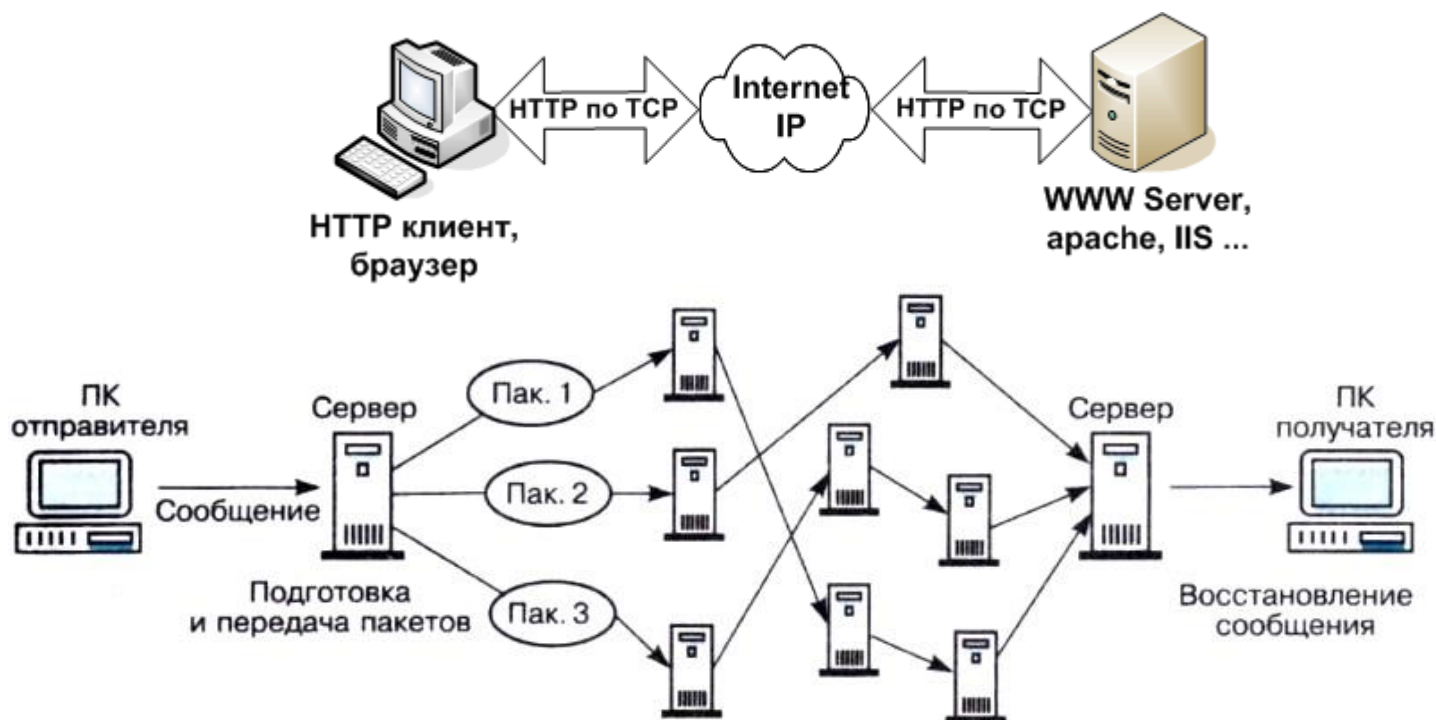


Рис. 2.1.Packetная технология передачи информации в глобальной сети

2 Стандарты WWW

Служба WWW (World Wide Web) - предназначена для обмена гипертекстовой информацией.

В начале служба WWW базировалась на трех стандартах:

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

Позже добавили:

- ✚ **CGI (Common Gateway Interface)** - универсальный интерфейс шлюзов. Создан для взаимодействия HTTP - сервера с другими программами, установленными на сервере (например, СУБД).

3 Принципы организации DNS и URL

Числовая IP-адресация неудобна для человека. Запомнить наборы цифр гораздо труднее, чем слова. Для облегчения стали использовать соответствия числовых адресов именам машин.

Например, для некоторого сервера **ks.ru** существуют следующие соответствия:

127.0.0.1 localhost - Длина имени дб не более 63 символов

Сначала такие соответствия просто сами записывали в файл, брали у знакомых и копировали с FTP-серверов.

Это файлы имеют название hosts, и находятся в каталогах:

- Для UNIX - /etc/hosts
- Для Windows - C:\windows\system32\drivers\etc\hosts

Можете сами изменить файл, например, внеся запись:

195.208.44.20 ipm

Теперь вы можете, просто набрав в браузере доменное имя ipm попасть на этот сервер www.ipm.ks.ru. Таким образом, вы можете попасть на любой сервер.

Однако такой способ присвоения символьных имен был хорош до тех пор, пока Internet был маленьким. По мере роста сети стало затруднительным поддерживать большие списки имен на каждом компьютере. Для того, чтобы решить эту проблему, были разработана служба DNS (Domain Name System).

Первый стандарт DNS определен в RFC0883 (Domain names: Implementation specification P.V. Mockapetris Nov-01-1983) и RFC0882 (Domain names: Concepts and facilities P.V. Mockapetris Nov-01-1983)

Последняя версия [RFC1035](#) (Domain names - implementation and specification P.V. Mockapetris Nov-01-1987)

Система доменных адресов строится по иерархическому принципу. Администрирование начинается с **доменов верхнего или первого уровня**.

Домен - определенная зона в системе доменных имён (DNS) Интернет, выделенная какой-либо стране, международной организации, региону, юридическому или физическому лицу для целей обеспечения доступа к предоставляемой им в Интернете информации.

Примеры доменов: .edu, .ru, .com, .org, .net и т.д.

Первые домены верхнего уровня были рассчитаны на США:

- gov - государственные организации
- mil - военные учреждения
- edu - образовательные учреждения
- com - коммерческие организации
- net - сетевые организации

Позднее, когда сеть перешагнула национальные границы США появились национальные домены типа:

- uk - Объединенное королевство
- jp - Япония
- au - Австралия
- ch - Чехия
- su – СССР
- ua- Украина
- ru – Россия и т.п.

В общей сложности на сегодняшний день насчитывается порядка 250 национальных доменов.

Доменное имя - в системе DNS символьная строка, состоящая из имени домена ([и субдоменов]) и имени хоста в этом домене, сопоставленная с сетевым адресом этого хоста.

Примеры доменных имен: donntu.org, example.com, test.example.com, test.test.example.com.

В настоящее время практически любая организация или физическое лицо может получить свой собственный домен второго уровня.

IANA - The Internet Assigned Numbers Authority (Управление назначением адресов в Internet) - организация, осуществляющая контроль над распределением доменов первого уровня. Сервер <http://www.iana.org/>.

Базу можно посмотреть по адресу через WWW-интерфейс - <http://whois.iana.org/>

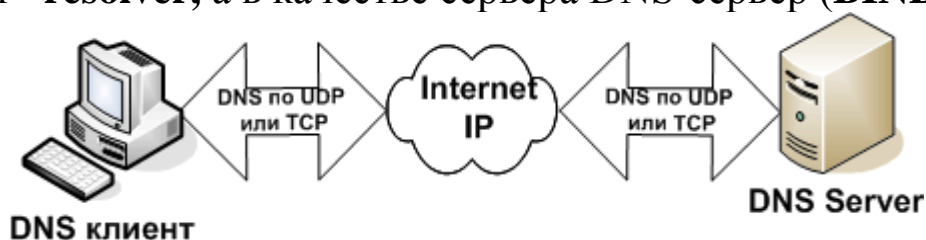
Вслед за доменами первого уровня следуют домены либо географические (yandex.ru), либо организации (http://donntu.org).

По итогам января 2017 года число зарегистрированных доменных имен в российских национальных доменах .ru выросло на 33 442 доменных имени и составило 6 355 685 доменных имен.

Далее идут домены третьего и более низких уровней, например: http://asus.cs.donntu.org

Служба доменных имен работает как распределенная база, данные которой распределены по DNS-серверам. Систему доменной адресации можно представить в виде дерева доменных имен.

Рассмотрим взаимодействие клиента и сервера по протоколу DNS (Алгоритм разрешения имен). Сервис DNS строится по схеме "клиент-сервер". В качестве клиентской части выступает процедура разрешения имен - **resolver**, а в качестве сервера DNS-сервер (**BIND** ...).



Например, когда мы хотим обратиться к серверу ipm.ks.ru, ваш браузер, используя **resolver**, поступает следующим образом:

1. ищет запись ipm.ks.ru в файле **hosts**, если не находит, то:
2. посылает запрос на известный DNS сервер (как правило, локальный), если на этом сервере запись не найдена, то,
3. сервер DNS обращается к DNS-ROOT серверу с запросом адреса DNS сервера, отвечающего за домен первого уровня ru, если получает адрес, то

4. сервер DNS обращается к DNS серверу, отвечающего за домен первого уровня ru, с запросом адреса DNS сервера отвечающего за домен второго уровня kstu.ru, если получает адрес, то,
5. сервер DNS посылает запрос на DNS сервер, отвечающий за домен второго уровня kstu.ru, если получает адрес, то,
6. сервер DNS адрес кэширует и передает клиенту
7. клиент обращается по IP адресу - 195.208.44.20

Для доступа к любым сетевым ресурсам необходимо знать, где они размещены и как к ним можно обратиться. Во Всемирной паутине используется стандартизованная схема адресации и идентификации, учитывающая опыт адресации и идентификации e-mail, Gopher, WAIS, telnet, ftp и т.п. - **URL, Uniform Resource Locator**.

URI (Uniform Resource Identifier, Универсальный идентификатор ресурса) (RFC 2396, August 1998) - компактная строка символов для идентификации абстрактного или физического ресурса. Под ресурсом понимается любой объект, принадлежащий некоторому пространству. Включает и переопределяет определенные ранее **URL** (RFC 1738/RFC 1808) и **URN** (RFC 2141, RFC 2611).

URI предназначен для уникальной идентификации любого ресурса.

А) Некоторые подмножества URI:

URN (Uniform Resource Name, Универсальное имя ресурса) - частная URI-схема.

Синтаксис: **urn: namespace: data1.data2, more-data,**

где namespace (пространство имен) определяет, каким образом используются данные, указанные после второго ":".

Пример URN: **urn: ISBN: 0-395-36341-6**

ISBN - тематический классификатор для издательств

0-395-36341-6 - конкретный номер тематики книги или журнала

При получении URN клиентская программа обращается к ISBN (каталогу "тематический классификатор для издательств" в Интернете). И получает расшифровку номера тематики "0-395-36341-6" (например: "квантовая химия").

URN массово используется в P2P сетях (подобных edonkey).

Пример URN, указывающего на образ диска Adobe Photoshop v8.0 в сети edonkey:

urn:ed2k://file|AdobePhotoshopv8.0.iso|940769280|b34c101c90b6dedb4071094cb1b9f2d3|/

где:

ed2k - указывает на сеть

file - файл

Adobe Photoshop v8.0.iso - название файла

940769280 - размер в байтах

b34c101c90b6dedb4071094cb1b9f2d3 - идентификатор файла (вычисляется с помощью хеш-функции)

б) Универсальный указатель ресурса URL:

URL (Uniform Resource Locator, [RFC 1738](#)) - унифицированный локатор (указатель) ресурсов, стандартизированный способ записи адреса ресурса в WWW и сети Интернет. Адрес URL имеет гибкую и расширяемую структуру для максимально естественного указания местонахождения ресурсов в сети, который идентифицирует ресурс по способу доступа к нему (например, его "местонахождению в сети") вместо того, чтобы идентифицировать его по названию или другим атрибутам этого ресурса.

URL - Uniform Resource Locators явно описывает, как добраться до объекта.

Примеры URL:

<http://www.ipm.ks.ru/index.php>

<ftp://www.ipm.ks.ru/>

Для представления адреса используется ограниченный набор символов ASCII.

Общий вид адреса можно представить так:

<схема>://<логин>:<пароль>@<хост>:<порт>/<полный-путь-к-ресурсу>

Где:

схема обращения к ресурсу: HTTP, FTP, GOPHER, MAILTO, NEWS, TELNET, FILE, MAN, INFO, WHATIS, LDAP, WAIS и т.п.

логин: *пароль*- имя пользователя и его пароль, используемые для доступа к ресурсу

хост-доменное имя хоста или его IP-адрес.

Порт- порт хоста для подключения

полный-путь-к-ресурсу - уточняющая информация о месте нахождения ресурса (зависит от протокола).

Появление адресов URL стало существенным нововведением в Интернете. Однако с момента его изобретения и по сей день стандарт URL обладает серьёзным недостатком — в нём можно использовать только ограниченный набор символов, даже меньший, нежели в ASCII: латинские буквы, цифры и лишь некоторые знаки препинания- **[a-z0-9+.-]**.

Если мы захотим использовать в URL символы кириллицы, или иероглифы, или, скажем, специфические символы французского языка, то нужные нам символы должны быть перекодированы особым образом.

В русскоязычной Википедии ежедневно приходится видеть примеры кодирования URL, поскольку русский язык использует символы кириллицы. Например, строка вида:

<http://ru.wikipedia.org/wiki/Микрокредит>

кодируется в URL как:

<http://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D1%80%D0%B5%D0%B4%D0%B8%D1%82>

Такое преобразование происходит в два этапа: сначала каждый символ кириллицы кодируется в Юникоде (UTF-8) в последовательность из двух байтов, а затем каждый байт этой последовательности записывается в шестнадцатеричном представлении:

М → D0 и 9C → %D0%9C

и → D0 и B8 → %D0%B8

к → D0 и BA → %D0%BA

р → D1 и 80 → %D1%80, и т. д.

Перед каждым таким шестнадцатеричным кодом байта, согласно спецификации URL, ставится знак процента (%) — отсюда даже возник английский термин «percent-encoding», обозначающий способ кодирования символов в URL и URI.

Поскольку такому преобразованию подвергаются буквы всех алфавитов, кроме базовой латиницы, то URL со словами на подавляющем большинстве языков (кроме английского, итальянского, латинского) может стать нечитаемым для человека.

Это всё входит в противоречие с принципом интернационализма, провозглашаемого всеми ведущими организациями Интернета, включая W3C. Эту проблему призван решить стандарт IRI (International Resource Identifier) — международных идентификаторов ресурсов, в которых можно было бы без проблем использовать символы Юникода, и которые поэтому не ущемляли бы права других языков.

Другие схемы URL

в)Схема HTTP

В схеме указывается ее идентификатор, адрес машины, TCP-порт, путь в директории сервера, переменные и их значения, метка.

Синтаксис:

http://[<user>[:<password>]>@]<host>[:<port>][/<url-path>][?<query>]]

http - название схемы

user - имя пользователя

password - пароль пользователя

host - имя хоста

port - номер порта

url-path - путь к файлу и сам файл

query (<имя-поля>=<значение>{&<имя-поля>=<значение>}) - строка запроса

Определен в RFC 2068. По умолчанию, port=80.

Примеры:

<http://ipm.ks.ru/internet/index.php>

Это наиболее распространенный вид URI, применяемый в документах WWW. Вслед за именем схемы (http) следует путь, состоящий из доменного адреса машины и полного адреса HTML-документа в дереве сервера HTTP.

В качестве адреса машины допустимо использование и IP-адреса:

<http://195.208.44.20/internet/index.php>

Если сервер протокола HTTP запущен на другой, отличный от 80 порт TCP, то это отражается в адресе:

<http://195.208.44.20:8080/internet/index.php>

При указании адреса ресурса возможна ссылка на метку внутри файла HTML. Для этого вслед за именем документа может быть указана метка внутри документа:

<http://195.208.44.20/internet/index.php#metka1>

Символ "#" отделяет имя документа от имени метки.

Переменные и их значения передаются следующим образом:

<http://ipm.kstu.ru/internet/index.php?var1=value1&vard2=value2>

Значения "var1" и "var2" - это имена переменных, а "value1" и "value2" - их значения.

[#запрос стартовой страницы по умолчанию](http://example.com)

[#запрос заданной страницы в указанном каталоге](http://www.example.com/site/map.html)

[#подключение на нестандартный порт](http://example.com:81/script.php)

[#запрос с передачей параметров скрипту](http://example.org/script.php?key=value)

[#подключение по сетевому адресу](http://192.168.0.1/example/www)

г)Схема FTP

Данная схема позволяет адресовать файловые архивы FTP.

Синтаксис:

ftp://[<user>[:<password>]>@]<host>[:<port>][/<url-path>]

ftp - название схемы

user - имя пользователя

password - пароль пользователя

host - имя хоста

port - номер порта

url-path - путь к файлу и сам файл

Определен в RFC 1738. По умолчанию, port=21, user=anonymous, password=email-адрес, если имя указано, а пароль нет, то он запрашивается в диалоге.

<url-path> имеет вид:

<cwd1>/<cwd2>/.../<cwdN>/<name>[:type=<typecode>], где <typecode>:

<url-path>

Примеры: ftp://ipm.kstu.ru/students/name/

Чтобы указать имя пользователя и его пароль, надо записать так:
ftp://name:password@ftp://ipm.kstu.ru/students/name/

В данном случае эти параметры отделены от адреса машины символом "@", а друг от друга двоеточием.

д)Схема MAILTO

Данная схема предназначена для отправки почты.

Синтаксис:

mailto:[<e-mail-1>{,<e-mail-2>,...}][?<query>]

mailto - название схемы

e-mail-1 (<user>@<host>)- первый адрес электронной почты

user - имя пользователя

host - имя хоста

e-mail-2 - второй адрес электронной почты

query (<имя-поля-заголовка>=<значение> {&<имя-поля-заголовка>=<значение> }) - строка запроса

Примеры:

mailto:name@ipm.kstu.ru

В этой схеме передаются поля и их значения:

Пример:

mailto:name@ipm.kstu.ru?subject=Тема_письма&body=Текст_который
_будет_вставлен_в_письмо

Адрес получателя можно также записывать в виде значения поля to:

mailto:?to=name@ipm.kstu.ru?subject=Тема_письма&body=Текст_которы
й_будет_вставлен_в_письмо

4. Описание протокола HTTP

Hypertext Transfer Protocol - протокол передачи гипертекста, протокол высокого уровня (а именно, уровня приложений). Используется службой WWW для передачи Web-страниц.

Первый документ (но не стандарт) - RFC1945 (Hypertext Transfer Protocol -- HTTP/1.0 T. Berners-Lee, R. Fielding, H. Frystyk May 1996)

Последняя версия - [RFC2616](#) (Hypertext Transfer Protocol -- HTTP/1.1 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee June 1999)

HTTP - типичный клиент-серверный протокол, обмен сообщениями идёт по схеме «запрос-ответ» в виде ASCII-команд. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Протокол HTTP определяет запрос-ответный способ взаимодействия между программой-клиентом и программой-сервером в рамках технологии World Wide Web.

Для загрузки Web-страницы в клиентский браузер тот посылает установленной на серверном компьютере специальной программе, называемой http-сервером, соответствующий запрос и обрабатывает полученные от него данные. В данном случае функции браузера состоят в том, чтобы запросить у сервера определенную страницу, получить ее и отобразить на экране пользователя. Сервер же принимает запрос, ищет запрошенный документ и выдает клиенту либо содержимое найденного файла, либо сообщение об ошибке, если такой файл не был найден или доступ к нему почему-либо запрещен. Важным моментом для понимания данного процесса является то, что http-сервер не анализирует содержимое передаваемого документа.

Поиск запрашиваемой страницы осуществляется в определенной директории, которая отведена на серверном компьютере под данный сайт — ссылка на эту директорию присутствует во введенном пользователем адресе. В случае, когда обращение осуществляется не к конкретному документу, а к сайту в целом, http-сервер автоматически подставляет вместо названия передаваемого файла так называемую «стартовую страницу», которая носит имя index.htm или index.html (в некоторых случаях — default.htm или default.html). Этот документ обязательно должен располагаться в корневом каталоге, отведенном для размещения вашего сайта, либо, если это оговорено особо, в директории с названием WWW. Все остальные файлы можно размещать либо в этом же каталоге, либо во вложенных директориях, что иногда бывает удобным, особенно в случае, когда сайт содержит несколько тематических разделов или рубрик.

Следует также помнить, что большинство серверных программ (в частности, Apache для UNIX-совместимых платформ) различают строчный и заглавный регистр символов, поэтому все имена файлов и их расширения во избежание ошибок следует писать строчными буквами, причем обязательно латиницей. Последнее связано с различиями в обработке кодировок русского языка, характерной для тех или иных серверов.

Работа по протоколу HTTP происходит следующим образом: программа-клиент устанавливает TCP-соединение с сервером (стандартный номер порта-80) и выдает ему HTTP-запрос. Сервер обрабатывает этот запрос и выдает HTTP-ответ клиенту.

Сообщения HTTP делятся на запросы клиента серверу и ответы сервера клиенту.

Сообщения запроса и ответа имеют общий формат. Оба типа сообщений выглядят следующим образом: сначала идет начальная строка (start-line), затем, возможно, одно или несколько полей заголовка, называемых, также, просто заголовками, затем пустая строка (то есть строка, состоящая из

символов CR и LF), указывающая конец полей заголовка, а затем, возможно, тело сообщения:

```
начальная строка
поле заголовка 1
поле заголовка 2
. . .
поле заголовка N
CR LF
тело сообщения
```

а) Заголовки протокола HTTP

Формат начальной строки клиента и сервера различаются и будут рассмотрены ниже. Заголовки бывают четырёх видов:

- общие заголовки (general-headers), которые могут присутствовать как в запросе, так и в ответе;
- заголовки запросов (request-headers), которые могут присутствовать только в запросе;
- заголовки ответов (response-headers), которые могут присутствовать только в ответе;
- заголовки объекта (entity-headers), которые относятся к телу сообщения и описывают его содержимое.

Каждый заголовок состоит из названия, символа двоеточия ":" и значения. Наиболее важные заголовки приведены в таблице 1.

Таблица 1

Заголовки протокола HTTP

Заголовок	Назначение
Заголовки объекта	
Allow	Перечисляет поддерживаемые сервером методы
Content-Encoding	Способ, которым закодировано тело сообщения, например, с целью уменьшения размера
Content-Length	Длина сообщения в байтах
Content-Type	Содержит обозначение типа содержимого ответа в MIME. Некоторые типы содержимого: text/html - текст в формате HTML (веб-страница); text/plain - простой текст (аналогичен "блокнотовскому"); image/jpeg - картинка в формате JPEG; image/gif - то же, в формате GIF;
Expires	Дата и время, когда ресурс на сервере будет изменён, и его нужно получать заново
Last-Modified	Дата и время последней модификации содержимого

Заголовки ответа	
Age	Число секунд, через которое нужно повторить запрос для получения нового содержимого
Location	URI ресурса, к которому нужно обратиться для получения содержимого
Retry-After	Дата и время или число секунд, через которое нужно повторить запрос, чтобы получить успешный ответ
Server	Название программного обеспечения сервера, приславшего ответ
Заголовки запроса	
Accept	Список поддерживаемых браузером типов содержимого в порядке их предпочтения данным браузером, например: Accept: image/gif, image/x-xbitmap, image/jpeg
Accept-Charset	Кодировки символов, в которых клиент может принимать текстовое содержимое: windows-1251, koi8-rus
Accept-Encoding	Способ, которым сервер может закодировать сообщение
Host	Хост и номер порта, с которого запрашивается документ
If-Modified-Since If-Match If-Range и др	Заголовки запроса для условного обращения к ресурсу
Range	Запрос части документа
User-Agent	Название программного обеспечения клиента- значением является "кодовое обозначение" браузера, например: Mozilla/4.0 (compatible; MSIE 5.0; Windows 95; DigExt)
Общие заголовки	
Connection	Connection (соединение) - может принимать значения Keep-Alive и Close. Keep-Alive означает, что после выдачи данного документа соединение с сервером не разрывается, и можно выдавать еще запросы. Большинство браузеров работают именно в режиме Keep-Alive, так как он позволяет за одно соединение с сервером "скачать" html-страницу и рисунки к ней. Будучи однажды установленным, режим Keep-Alive сохраняется до первой ошибки или до явного указания в очередном запросе Connection: Close. - соединение закрывается после ответа на данный запрос.
Date	Дата и время формирования сообщения
Pragma	Специальные, зависящие от реализации команды, касающиеся передаваемого содержимого
Transfer-Encoding	Способ кодирования сообщения при передаче

В некоторых заголовках значением является дата и время. Они должны быть представлены в формате, описанном в RFC 1123, например:

Sun, 06 Nov 1994 08:49:37 GMT

В теле сообщения содержится собственно передаваемая информация – полезная нагрузка сообщения. Тело сообщения представляет собой последовательность октетов (байтов).

Сообщение запроса от клиента к серверу состоит из строки запроса (request-line), заголовков (общих, запросов, объекта) и, возможно, тела сообщения.

Строка запроса начинается с метода, затем следует идентификатор запрашиваемого ресурса, версия протокола и завершающие символы конца строки:

<Метод> <Идентификатор> <Версия HTTP>

Метод указывает метод, который нужно применить к запрашиваемому ресурсу. Например, метод GET говорит о том, что клиент хочет получить содержимое ресурса. Идентификатор определяет запрашиваемый ресурс. Версия HTTP обозначается строкой следующего вида:
HTTP/<версия>.<подверсия>

Б) Методы протокола HTTP

Рассмотрим основные методы протокола HTTP.

OPTIONS

Метод OPTIONS выполняет запрос информации об опциях соединения (например, методах, типах документов, кодировках), которые поддерживает сервер для запрашиваемого ресурса. Этот метод позволяет клиенту определять опции и/или требования, связанные с ресурсом, или возможности сервера, не производя никаких действий над ресурсом и не иницируя его загрузку.

GET

Метод GET позволяет получать любую информацию, связанную с запрашиваемым ресурсом. В большинстве случаев, если идентификатор запрашиваемого ресурса указывает на документ (например, текстовый документ, графическое изображение, видеоролик), то сервер возвращает содержимое этого документа (содержимое файла). Если запрашиваемый ресурс является приложением (программой), формирующим данные, то в теле сообщения ответа возвращаются сформированные данные, а не двоичный образ выполняемого файла. Это используется, например, при создании приложений CGI. Если идентификатор запрашиваемого ресурса указывает на директорию (каталог, папку), то, в зависимости от настроек сервера, может быть возвращено либо содержимое директории (список файлов), либо содержимое одного из файлов, находящегося в этой директории (как правило, index.html или Default.htm). В последнем случае имя папки может указываться как с символом "/" на конце, так без него. При отсутствии на конце

идентификатора данного символа сервер выдаёт один из ответов с перенаправлением (с кодами статуса 301 или 302).

Различается "условный GET" ("conditional GET"), при котором сообщение запроса включает заголовки запроса If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, или If-Range. Условный метод GET запрашивает передачу объекта, только если он удовлетворяет условиям, описанным в приведённых заголовках. Условный метод GET предназначен для уменьшения ненужной загрузки сети, поскольку позволяет не загружать вторично уже сохранённые клиентом данные.

Различается также "частичный GET" ("partial GET"), при котором сообщение запроса включает заголовок запроса Range. Частичный GET запрашивает передачу только части объекта. Частичный метод GET предназначен для уменьшения ненужной загрузки сети, за счёт запроса только части объекта, когда другая часть уже загружена клиентом. Значением заголовка Range является диапазон байтов, которые необходимо получить. Байты нумеруются с 0. Начальный и конечный байты диапазона разделяются символом "-". Если нужно получить несколько диапазонов, то они перечисляются через запятую.

HEAD

Метод HEAD идентичен GET, за исключением того, что сервер не возвращает в ответе тело сообщения. Метаинформация, содержащаяся в HTTP заголовках ответа на запрос HEAD идентична информации, представляемой в ответ на запрос GET. Этот метод может использоваться для получения информации об объекте запроса без непосредственной пересылки тела объекта. Метод HEAD часто используется для тестирования гипертекстовых связей.

POST

Метод POST используется для запроса, при котором адресуемый сервер принимает данные, включённый в тело сообщения (объект) запроса, и отправляет их на обработку приложению, указанному как запрашиваемый ресурс. POST разработан для того, чтобы общим методом реализовать следующие функции:

- аннотация существующих ресурсов;
- регистрация сообщения на электронной доске объявлений (BBS), в конференциях новостей (newsgroups), списках рассылки (mailing lists) или подобной группе статей;
- передача блока данных, например результат ввода в форме, процессу обработки;
- выполнение запросов к базам данных (БД);
- загрузка файлов на сервер.

Браузер может формировать запросы с методом POST при отправке форм. Для этого элемент FORM документа HTML, содержащего форму, должен иметь атрибут METHOD со значением POST.

Действие, выполняемое методом POST, может выполнить действие на сервере и не передать никакого содержимого в качестве результата работы. В этом случае, в зависимости от того, включает ответ тело сообщения, описывающее результат, или нет, код состояния в ответе может быть как 200 (OK), так и 204 (Нет содержимого, No Content).

Если ресурс на сервере был создан, ответ содержит код состояния 201 (Создан, Created) и включает заголовок ответа Location.

PUT

Тело сообщения, которое передаётся в запросе с методом PUT, сохраняется на сервере, причём идентификатор запрашиваемого ресурса будет идентификатором сохранённого документа. Фундаментальное различие между методами POST и PUT заключается в различном значении идентификатора запрашиваемого ресурса. URI в запросе POST идентифицирует ресурс, который обрабатывает включенный в тело сообщения объект. Этим ресурсом может быть приложение, принимающее данные. Напротив, URI в запросе PUT идентифицирует объект, включенный в запрос в виде тела сообщения, то есть пользовательский агент назначает данный URI включенному ресурсу.

DELETE

Метод DELETE запрашивает сервер об удалении ресурса, имеющего запрашиваемый идентификатор. Запрос с данным методом может быть отвергнут сервером, если у пользователя нет прав на удаление запрашиваемого ресурса.

TRACE

Метод TRACE используется для возврата переданного запроса на уровне протокола HTTP. Получатель запроса (сервер Web) отправляет полученное сообщение обратно клиенту как тело объекта ответа с кодом состояния 200 (OK). Запрос TRACE не должен содержать тела сообщения.

TRACE позволяет клиенту видеть, что получает на другом конце сервер и использовать эти данные для тестирования или диагностики.

Если запрос успешно выполнен, то ответ содержит всё сообщение запроса в теле сообщения ответа, а заголовок объекта Content-Type имеет значение "message/http".

В) Коды ответов

После получения и интерпретации сообщения запроса, сервер отвечает сообщением HTTP ответа.

Первая строка ответа – это строка состояния (Status-Line). Она состоит из версии протокола, числового кода состояния, поясняющей фразы, разделенных пробелами и завершающих символов конца строки:

<Версия HTTP> <Код состояния> <Поясняющая фраза>

Версия протокола имеет то же значение, что и в запросе.

Элемент код состояния (Status-Code) - это целочисленный трехразрядный (трёхзначный) код результата понимания и удовлетворения запроса. Поясняющая фраза (Reason-Phrase) представляет собой короткое текстовое описание кода состояния. Код состояния предназначен для обработки программным обеспечением, а поясняющая фраза предназначена для пользователей.

Первая цифра кода состояния определяет класс ответа. Последние две цифры не имеют определенной роли в классификации. Имеется 5 значений первой цифры:

- 1xx: Информационные коды - запрос получен, продолжается обработка.
- 2xx: Успешные коды - действие было успешно получено, понято и обработано.
- 3xx: Коды перенаправления - для выполнения запроса должны быть предприняты дальнейшие действия.
- 4xx: Коды ошибок клиента - запрос имеет ошибку синтаксиса или не может быть выполнен.
- 5xx: Коды ошибок сервера - сервер не в состоянии выполнить допустимый запрос.

Поясняющие фразы (Reason-Phrase) для каждого кода состояния перечислены в RFC 2068 и являются рекомендуемыми, но могут быть заменены на эквивалентные без воздействия на протокол. Например, в локализованных русскоязычных версиях HTTP серверов эти фразы заменены русскими. В таблице 2 приведены коды ответов сервера HTTP.

Таблица 2

Коды ответов сервера HTTP

Код	Поясняющая фраза согласно RFC 2068	Эквивалентная поясняющая фраза на русском языке
1xx: Информационные коды		
100	Continue	Продолжать
2xx: Успешные коды		
200	OK	ОК
201	Created	Создан
204	No Content	Нет содержимого
205	Reset Content	Сбросить содержимое
206	Partial Content	Частичное содержимое
3xx: Коды перенаправления		
302	Moved Temporarily	Временно перемещен
304	Not Modified	Не модифицирован

4xx: Коды ошибок клиента		
400	Bad Request	Испорченный Запрос
401	Unauthorized	Несанкционированно
404	Not Found	Не найден
405	Method Not Allowed	Метод не дозволен
408	Request Timeout	Истекло время ожидания запроса
409	Conflict	Конфликт
411	Length Required	Требуется длина
413	Request Entity Too Large	Объект запроса слишком большой
5xx: Коды ошибок сервера		
500	Internal Server Error	Внутренняя ошибка сервера
501	Not Implemented	Не реализовано
503	Service Unavailable	Сервис недоступен
505	HTTP Version Not Supported	Не поддерживаемая версия HTTP

За строкой состояния следуют заголовки (общие, ответа и объекта) и, возможно, тело сообщения.

Рассмотрим простейший пример HTTP - запроса.

Если в адресном окне браузера мы наберем адрес **http://yandex.ru**, то браузер определит IP адрес сервера yandex.ru и пошлет ему на 80-й порт такой HTTP запрос:

GET http://yandex.ru/ HTTP/1.0

*Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */**

Accept-Language: ru

Cookie: yandexuid=2464977781018373381

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)

Host: yandex.ru

Referer: narod.ru

Proxy-Connection: Keep-Alive

Запрос передается в незашифрованном текстовом виде.

Самая главная часть запроса расположена в первой строке.

Это тип запроса (GET), URL адрес запрашиваемого документа(http://yandex.ru) и версия HTTP протокола (HTTP/1.0). Далее перечисляются параметры запроса. Каждая строка соответствует одному параметру.

В начале строки идет имя параметра, затем двоеточие и значение параметра.

Ассерп - тип данных, которые может принять браузер (в кодировке MIME).

Accept-Language - предпочтительный язык, на котором браузер хочет принять данные. User-Agent - тип программы, которая отослала запрос.

Host – DNS (или IP) имя хоста, которому адресован запрос.

Cookie - куки (данные, которые были сохранены сервером на локальном диске клиента, при посещении данного хоста прошлый раз).

Referer - хост, со страницы которого мы отсылаем запрос. Так, например, если мы находимся на странице <http://narod.ru>, и нажимаем там ссылку <http://yandex.ru>, то запрос будет отправлен хосту yandex.ru, а поле запроса referer будет содержать имя хоста narod.ru. Набор параметров запроса не фиксирован. Помимо приведенных, могут присутствовать и другие параметры.

Запрос GET может содержать данные, передаваемые клиентом серверу. Они передаются непосредственно через URL адрес по CGI протоколу. Данные отделены от URL знаком “?” и соединяются знаком “&”:

GET <URL>?<параметр 1>=<значение 1>&<параметр 2>=<значение 2>&...

Такой тип передачи данных серверу удобен, однако имеет ограничения на объем. Слишком большие массивы данных передавать через URL нельзя. Для таких целей существует другой тип запросов: запрос POST. Запрос POST очень похож на GET, с той лишь разницей, что данные в запросе POST передаются отдельно от самого заголовка запроса.

Тело запроса должно отделяться от заголовка пустой строкой. Если сервер встречает пустую строку в POST запросе, то все что идет далее он считает телом запроса (передаваемыми данными). Отметим следующее: формат данных в теле POST запроса произволен. Кроме того, POST запрос не требует наличия тела запроса, и может передавать данные также и через URL.

Современные браузеры содержат инструменты для веб-разработчиков, позволяющие получить некоторую информацию об отправляемых запросах POST. Если вам требуется посмотреть заголовки всего пары запросов, их использование будет проще и быстрее других способов.

Если вы используете Firefox, вы можете воспользоваться его веб-консолью. Она отображает заголовки запросов и содержимое передаваемых файлов cookies. Для ее запуска раскройте меню браузера, щелкните по пункту «Веб-разработка» и выберите «Веб-консоль». В появившейся панели активируйте кнопку «Сеть». Введите в поле фильтра название метода – POST. В зависимости от ваших целей, нажмите на кнопку формы отправляющей нужный запрос или обновите страницу. В консоли отобразится отправленный запрос. Кликните по нему мышкой, чтобы посмотреть подробнее.

Браузер Google Chrome имеет мощные инструменты отладки. Чтобы ими воспользоваться, кликните по иконке с изображением гаечного ключа, а затем раскройте пункт «Настройка и управление Google Chrome». Выберите пункт

«Инструменты» и запустите «Инструменты разработчиков». В панели инструментов выберите вкладку Network и отправьте запрос. Найдите нужный запрос в списке и кликните по нему, чтобы изучить подробности.

В браузере Opera имеются встроенные инструменты для разработчиков Opera Dragonfly. Для их запуска кликните правой клавишей мышки по нужной странице и выберите пункт контекстного меню «Проинспектировать элемент». Перейдите на вкладку «Сеть» инструментов для разработчиков и отправьте нужный запрос. Найдите его в списке и раскройте, чтобы изучить заголовки и ответы сервера.

Internet Explorer 9 содержит комплект под названием «Средства разработчика F12», предоставляющие подробную информацию по выполненным запросам. Чтобы посмотреть запрос, перейдите на вкладку «Сеть». Найдите заданный запрос в сводке и с помощью двойного клика раскройте подробную информацию.

Браузеры Chrome и Internet Explorer 9 содержат встроенные инструменты, позволяющие изучить отправленный запрос post во всех деталях. Для получения полной информации используйте их или Firefox с установленным плагином Firebug. Он очень удобен для частого изучения запросов, например, при отладке сайтов.

Если вы хотите посмотреть запрос, отправленный не браузером, а какой либо другой программой, воспользуйтесь HTTP-отладчиком Fiddler. Он работает как прокси-сервер и перехватывает запросы любой программы, а также предоставляет очень подробную информацию по их заголовкам и содержанию.

5 Стандарт CGI

Common Gateway Interface (CGI) - это механизм доступа к программам на стороне сервера.

Спецификация Common Gateway Interface (CGI) была разработана для расширения возможностей WWW за счет подключения различного внешнего программного обеспечения. При использовании CGI веб-сервер представляет браузеру доступ к исполнимым программам, запускаемым на его (серверной стороне) через стандартные потоки ввода и вывода, что упрощает программирование до предела.

Механизм CGI можно разделить на четыре части:

1. переменные окружения, через них приложению передаются данные и служебная информация
2. командная строка (при запросах типа ISINDEX)
3. стандартный ввод, используемый при передаче данных по методу POST
4. стандартный вывод, который предназначен для возврата данных серверу.

Интерфейс CGI используется для создания динамических веб-сайтов, например, когда веб-страницы формируются из результатов запроса к базе данных. Сейчас популярность CGI снизилась, т.к. появились более совершенные альтернативные решения (например, модули расширения веб-серверов).

С помощью CGI-сценариев могут приниматься данные от пользователя, они позволяют организовать диалог на Web-страницах, запросы к базам данных и т.д. Создать CGI-сценарий можно: Perl, Basic, C, C++ и т.п.

6 Web-технологии

SSI - (Server Side Includes) — технология, тесно переплетенная с упомянутой выше CGI. На основе макроязыка, очень напоминающего C, SSI позволяет реализовать такие возможности, как вывод в документе того или иного текста в зависимости от определенных условий или согласно заданному алгоритму, формировать файл HTML из динамически изменяющихся фрагментов или встраивать результат работы CGI в какой-либо его участок. Достоинства и недостатки SSI аналогичны описанным в предыдущем абзаце.

ASP - (Active Server Pages, активные страницы сервера) — еще одна технология, аналогичная JavaScript и PHP. Для того чтобы сделать web-страницу интерактивной с применением технологии ASP, необходимо встроить в ее код соответствующий скрипт, написанный на макроязыке, отдаленно напоминающем Java и C. Скрипт интерпретируется и исполняется непосредственно на сервере, после чего пользовательскому браузеру отправляется уже готовый html-документ с результатами работы сценария ASP. Отсюда следует вполне справедливое заключение о том, что для страниц, содержащих ASP, не имеет значения, какое программное обеспечение установлено на пользовательском компьютере. Зато принципиальное значение имеет тип сервера, на котором вы планируете использовать ASP, поскольку отнюдь не все они поддерживают данную технологию.

Macromedia Flash был разработан компанией Macromedia в 1996 году. Основное назначение данной технологии — создание высококачественной интерактивной анимации, которую можно представлять при относительно небольшом размере итогового файла. С помощью Macromedia Flash web-мастер имеет возможность изготавливать красочные анимационные заставки, определенные элементы которых могут «реагировать» на движения мыши, а также встроенные в web-страницы мини-игры, озвученные мультипликационные клипы и многое другое. Другими словами, Macromedia Flash способна практически на все, что доступно при применении Java, и даже больше.

Одним из основных отличий Macromedia Flash от всех остальных существующих ныне web-технологий является то, что это единственный стандарт, позволяющий работать не только с растровой графикой, но и с векторной. Помимо этого, текст импортируется в среду Flash не как растровое

изображение, а именно как текст, в котором один символ кодируется одним байтом информации, что является серьезным оружием в борьбе с излишним объемом результирующих файлов. Немаловажно, что во Flash имеется возможность воспроизведения музыки и звуков в формате MP3.

Для создания документов, подготовленных согласно данному стандарту, компания Macromedia выпустила специальную объектно-ориентированную среду разработчика в виде стандартного 32-разрядного приложения Microsoft Windows. Благодаря этому собственно процесс программирования при подготовке интерактивных элементов Flash и анимации сведен к минимуму: пользователю достаточно создать новый проект в удобной и дружелюбной среде, для чего необходимо лишь разобраться в принципах работы программного пакета и уметь обращаться с мышью, — по окончании работы программа встроит созданные объекты в документ HTML автоматически.

В среду Flash интегрирован специальный язык программирования, немного похожий на Java, но более простой в изучении. Написанные с его помощью программные модули импортируются в документ как апплеты и вставляются в нужный кадр анимации, где должно произойти динамическое изменение изображения. С помощью специального редактора можно написать небольшую программку, управляющую проигрыванием клипа, создать элементы, поддающиеся индивидуальным настройкам посетителями сайта, генерировать заставку с несколькими вариантами продолжения. Способов реализации этой возможности существует множество, но все-таки, чтобы использовать всю мощь встроенного языка, необходимо иметь определенный опыт в программировании. Механизм технологии Macromedia Flash достаточно прост: в общем случае подготовленные разработчиком интерактивные элементы и анимация превращаются в интерпретируемый код, который импортируется в страницу HTML. Для того чтобы клиентский браузер мог распознать объекты Flash, необходимо проинсталлировать на компьютере пользователя специальную надстройку к браузеру (plug-in), которая носит название Macromedia Flash Player. Эта программа распространяется бесплатно фирмой-разработчиком данной технологии. Пользовательский браузер, сталкиваясь с документом в формате Flash, как правило, сам связывается с соответствующим узлом, после чего начинает загрузку и установку Flash Player в автоматическом режиме. Проигрыватель Flash уже интегрирован в браузеры Microsoft Internet Explorer. Другие браузеры, к сожалению, не умеют распознавать данный стандарт, в чем и заключается его основной недостаток.

JAVA

Язык программирования Java, разработанный компанией Sun Microsystems и напоминающий по структуре и синтаксису хорошо знакомый многим программистам C. Не стоит путать со скриптовым языком JavaScript, который является всего лишь надстройкой стандарта HTML и значительно

расширяет возможности документа, созданного в этом формате. Модуль, написанный на JavaScript, интегрируется в файл HTML как подпрограмма и вызывается на исполнение из соответствующей строки HTML-кода стандартной командой. Встроенный в браузер интерпретатор языка воспринимает и скрипт, и сам код гипертекста как единый документ, обрабатывая те и другие данные одновременно.

Модуль Java в отличие от JavaScript не интегрируется в использующую его страницу, а существует как самостоятельное приложение с расширением .class или, выражаясь научным языком, апплет. При использовании этого варианта языка апплет также вызывается из html-файла соответствующей командой, но загружается, инициализируется и запускается на исполнение в виде отдельной программы, в фоновом режиме.

С помощью технологии Java можно придать своей странице элементы интерактивности, формировать, компоновать и полностью контролировать формат всплывающих окон и встроенных фреймов, организовывать такие активные элементы, как «часы», «бегущие строки» и иную анимацию, создать чат. Большинство web-камер, передающих на сайт «живое» изображение, также работают на базе соответствующих приложений Java.

Среди достоинств этой технологии следует отметить отсутствие необходимости устанавливать и настраивать на сервере какие-либо дополнительные модули, обеспечивающие работу Java-программ. Главный недостаток Java заключается в том, что пользователи браузеров старых версий, не поддерживающих компиляцию данного языка, воспринимать объекты, созданные при помощи Java не смогут.

7 Программное обеспечение WWW

Веб-сервер — это сетевое приложение, обслуживающее HTTP-запросы от клиентов, обычно веб-браузеров. Веб-сервер принимает запросы и возвращает ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными. Веб-серверы — основа Всемирной паутины.

Созданием программного обеспечения Web-серверов занимаются многие разработчики, но наибольшую популярность в WWW получили такие программные продукты, как Apache ([Apache Software Foundation](#)), IIS ([Microsoft](#)), QZHTTP (он же qq.com), Google Web Server (GWS, Google Inc.)

Apache - свободный Web-сервер с открытым исходным кодом, распространяется под совместимой с GPL лицензией. Apache уже многие годы является лидером по распространенности во Всемирной паутине в силу своей надежности, гибкости, масштабируемости и безопасности.

IIS (Internet Information Services) - проприетарный набор серверов для нескольких служб Интернета, разработанный Майкрософт и распространяемый с ОС семейства Windows NT. Основным компонентом IIS

является веб-сервер, также поддерживаются протоколы FTP, POP3, SMTP, NNTP.

QZHTTP - модифицированный Apache, используемый на китайском портале qq.com. На нем размещены сервисы онлайн-дневников и мгновенного обмена сообщениями.

Google Web Server (GWS) - разработка компании Google на основе web-сервера Apache. GWS оптимизирован для выполнения приложений сервиса Google Applications.

nginx [engine x] — это HTTP-сервер, совмещенный с почтовым прокси-сервером. Разработан [И. Сысоевым](#) для компании Рамблер. Осенью 2004 года вышел первый публично доступный релиз, сейчас nginx используется рядом крупных сайтов.

lighttpd — веб-сервер, разрабатываемый с расчётом на быстроту и защищённость при использовании на сильно нагруженных сайтах, а также соответствие стандартам. lighttpd — свободное программное обеспечение, распространяемое по лицензии BSD.

Node.js -разработан в 2009 году Райаном Далем. В настоящее время проектом управляет Node.js Foundation при поддержке Linux Foundation(доля составляет 0,2% рынка). Node.js представляет собой платформу для выполнения сетевых приложений и зачастую играет роль самостоятельного веб-сервера. Благодаря своим конструктивным особенностям он позволяет добиться высокой пропускной способности и масштабируемости веб-приложений, работающих в режиме реального времени.

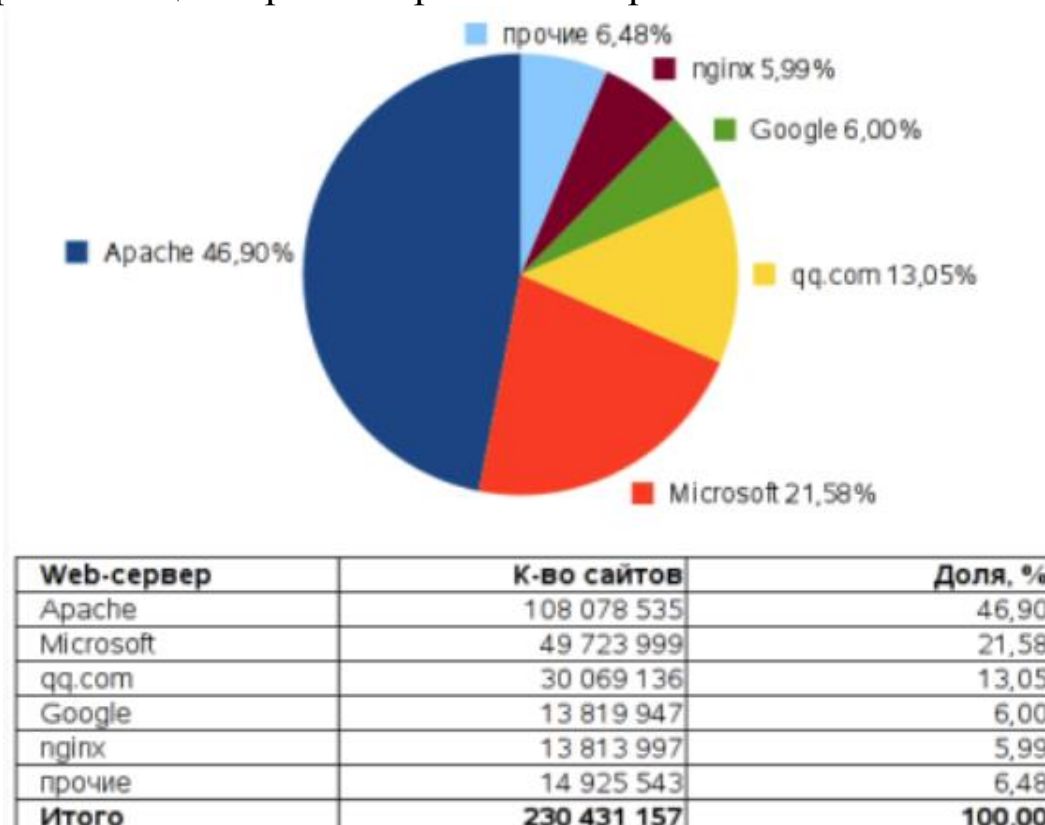
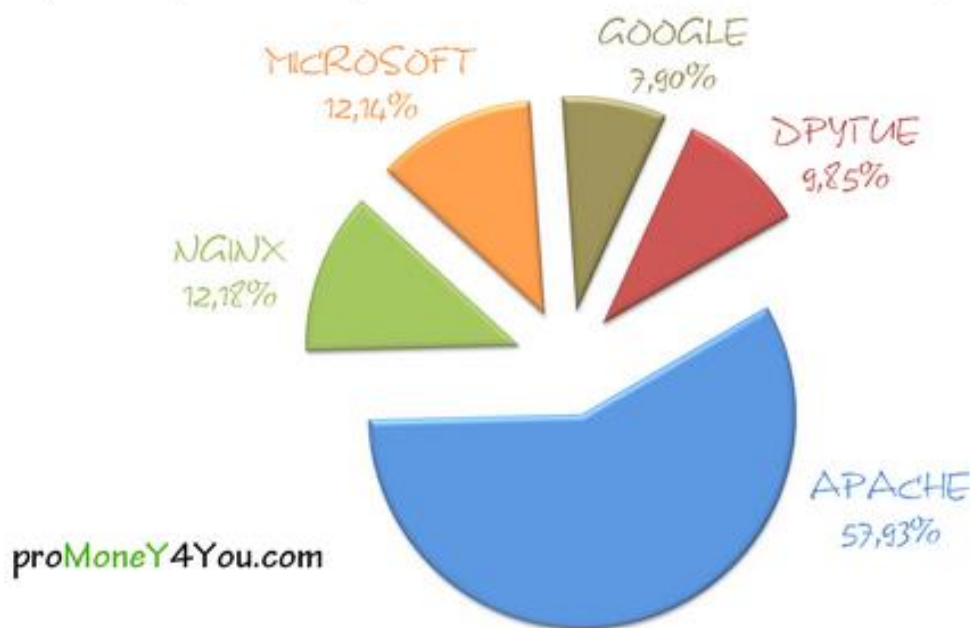


Рис. 2. Распределение web-серверов на октябрь 2009 г.(по данным [netcraft.com](#))

Самые популярные web-серверы
(январь 2012, только активные сайты)



8. Браузеры

Браузер, веб-обозреватель (web-browser) — это клиентское программное обеспечение для доступа к веб-серверам по протоколу HTTP и просмотра веб-страниц. Как правило браузеры дополнительно поддерживают и ряд других протоколов (например ftp, file, mms, pop3).

Первые HTTP-клиенты были консольными и работали в текстовом режиме, позволяя читать гипертекст и перемещаться по ссылкам. Сейчас консольные браузеры (такие, как lynx, w3m или links) практически не используются рядовыми посетителями веб-сайтов. Тем не менее такие браузеры весьма полезны для веб-разработчиков, так как позволяют "увидеть" веб-страницу "глазами" поискового робота.

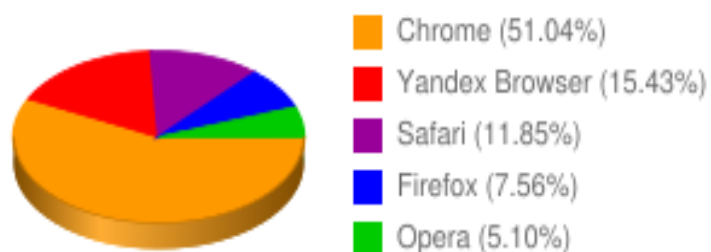
Исторически первым браузером в современном понимании (т.е. с графическим интерфейсом и т.д.) была программа NCSA Mosaic, разработанная Марком Андерисеном и Эриком Бина. Mosaic имел довольно ограниченные возможности, но его открытый исходный код стал основой для последующих разработок.

Ныне браузер — комплексное приложение для обработки и вывода разных компонентов веб-страницы.

Наибольшую распространенность в мире имеют четыре браузера (с появлением Google Chrome их стало пять): Internet Explorer (Microsoft), Firefox (Mozilla Foundation), Safari (Apple) и Opera (OperaSoftware ASA)

2017	Chrome	IE/Edge	Firefox	Safari	Opera
December	77.0 %	3.9 %	12.4 %	3.3 %	1.6 %

OpenStat



LiveInternet



Internet Explorer (IE) - браузер, разработанный компанией Майкрософт, интегрирован с ОС Windows и поставляется вместе с ней. Платформозависим (поддержка сторонних ОС прекращена, начиная с версии 5). Единственный браузер, напрямую поддерживающий технологию ActiveX. Не полностью совместим со стандартами W3C, в связи с чем требует дополнительных затрат от веб-разработчиков.

Firefox - свободный кроссплатформенный браузер, разрабатываемый Mozilla Foundation и распространяемый под тройной лицензией GPL/LGPL/MPL. В основе браузера - движок Gecko, который изначально создавался для Netscape Communicator. Однако, вместо того, чтобы предоставить все возможности движка в стандартной поставке, Firefox реализует лишь основную его функциональность, предоставляя пользователям возможность модифицировать браузер в соответствии с их требованиями через поддержку расширений (add-ons), тем оформления и плагинов.

Такая лёгкая расширяемость достигается, в основном, за счёт использования в интерфейсе специально разработанного исключительно для Gecko языка разметки XUL и стандартных JavaScript и CSS. Это иногда приводит к более медленной работе интерфейса и повышенным требованиям к оперативной памяти.

Safari - проприетарный браузер, разработанный корпорацией Apple и входящий в состав операционной системы Mac OS X. Бесплатно распространяется для операционных систем семейства Microsoft Windows. В браузере используется уникальный по производительности интерпретатор JavaScript, опережающий аналоги более чем в 2 раза. Среди других особенностей: функция «Snapback», которая позволяет мгновенно вернуться к исходным результатам поиска или к верхнему уровню любого веб-сайта; те же графические технологии, что и в Mac OS X; предварительная поддержка стандартов CSS3 и HTML5; автоматическое распознавание и подгрузка

отсутствующих в системе шрифтов; интеграция мультимедийных технологий QuickTime.

Google Chrome — браузер с открытым исходным кодом, разрабатываемый компанией Google. Первая стабильная версия вышла 11 декабря 2008 года. В отличие от многих других браузеров, в Chrome каждая вкладка является отдельным процессом. В случае если процесс обработки содержимого вкладки зависнет, его можно будет завершить без риска потери данных других вкладок. Еще одна особенность - интеллектуальная адресная строка (Omnibox). К возможности автозаполнения она добавляет поисковые функции с учетом популярности сайта, релевантности и пользовательских предпочтений (истории переходов). Планируется кроссплатформенность.

Opera — кроссплатформенный многофункциональный веб-браузер, разработанный в 1994 году группой исследователей из норвежской компании Telenor. Дальнейшая разработка ведется Opera Software ASA.

Браузер Opera обладает высокой скоростью работы и совместим с основными стандартами WWW. Отличительными особенностями Opera долгое время являлись многостраничный интерфейс (система вкладок) и возможность масштабирования веб-страниц целиком, вместе с графикой. Функция Small Screen Rendering позволяет переформатировать документ для отображения на экранах мобильных устройств. В Opera, кроме стандартных способов навигации предусмотрены так называемые «жесты мышью». На разных этапах развития в Opera были интегрированы возможности почтового/новостного клиента Opera Mail, адресной книги, клиента сети BitTorrent, агрегатора RSS, клиента IRC, менеджера загрузок, WAP-браузера, а также поддержка виджетов — графические модули, работающих вне окна браузера.

Специальные редакции, Opera Mobile и Opera Mini, существенно опережают прочие браузеры по числу установок на мобильные телефоны. Opera также единственный браузер для игровых систем NintendoDS и Wii. Движок Opera (Presto) лицензирован Adobe и интегрирован в пакет для веб-разработчиков Adobe Creative Suite.

С 2005 года версии Opera для ПК и устройств мобильной связи (кроме Opera Mobile) доступны бесплатно.

9 ТОП-10 поисковых систем мира

Поисковая система — программно-аппаратный комплекс с веб-интерфейсом, предоставляющий возможность поиска информации в Интернете

1. Google

Ожидается, что Гугл занял первое место в мировом рейтинге. На его долю приходится более 70% поисковых запросов от жителей со всех уголков Земли. Причем треть всего трафика google.com приходится на граждан США. Кроме того, Гугл является самым посещаемым сайтом в мире. Средняя ежедневная

продолжительность использования поисковой системы Гугл составляет 9 минут.

Преимуществом поисковика Гугл является отсутствие лишних элементов на странице. Лишь строка поиска и логотип компании. **Фишкой** являются анимированные картинки и браузерные игры, приуроченные к популярным и локальным праздникам.

2. Bing

Бинг — поисковая система от Microsoft, ведущая свою историю с 2009 года. С этого момента она стала обязательным атрибутом смартфонов на ОС Windows. Bing также отличает минимализм — помимо шапки с перечнем всех продуктов Microsoft, на странице расположена лишь поисковая строка и название системы. Наиболее популярен Бинг в США (31%), Китае (18%) и Германии (6%).

3. Yahoo!

Третье место закрепилось за одним из старейших поисковиков — Яху. Основная масса пользователей также проживает в США (24%). *Складывается впечатление, что весь остальной мир сознательно избегает помощи поисковых роботов...* Также поисковик популярен в Индии, Индонезии, Тайване и в Великобритании. Помимо строки поиска, на странице Yahoo! предлагается прогноз погоды в вашем регионе, а также мировые тренды в виде ленты новостей.

4. Baidu

Китайский поисковик, который в России снискал дурную славу. Из-за агрессивной политики и отсутствия перевода на русский или английский языки, расширения этой поисковой системы воспринимаются в качестве вирусов. Их очень сложно удалить до конца и избавиться от всплывающих окон с иероглифами. Тем не менее, этот сайт является четвертым в мире по посещаемости. 92% его аудитории составляют граждане Китая.

5. AOL

AOL — американская поисковая система, чье название расшифровывается как America Online. Ее популярность значительно ниже, чем у предыдущих систем. Ее расцвет приходился на 90-е и 00-е годы. Почти 70% аудитории АОЛ — жители Соединенных Штатов.

6. Ask.com

У этой поисковой системы, ведущей свою историю с 1995, довольно необычный интерфейс. Все запросы она воспринимает в качестве вопросов и предлагает варианты ответов в соответствии с поисковой выдачей. Это чем-то напоминает сервис Ответы. Мэйл. Однако в выдачу попадают не ответы любителей, а полноценные статьи. За последний год сайт потерял

около 50 позиций в мировом рейтинге самых популярных интернет-ресурсов и на сегодняшний день занимает лишь 104 место.

7. Excite

Этот поисковик ничем не примечателен, и похож на массу других сайтов. Он предлагает пользователям массу сервисов (таких, как Новости, Почта, Погода, Путешествия и т.д.) Интерфейс сайта также вызывает воспоминания о вэбе 90-х гг и, можно предположить, мало изменился с тех пор.

8. DuckDuckGo

Разработчики сразу предупреждают, что данная поисковая система не отслеживает ваши действия в сети. В наши дни это является весомым аргументом при выборе поисковика. Дизайн сайта выполнен в современной манере, использует яркие краски и забавные картинки. В отличие от других поисковых машин, «утиный поисковик» переведен на русский язык. За последний год сайт отыграл около 400 позиций и в марте 2017г. находится на 504 строчке рейтинга популярности Alexa.

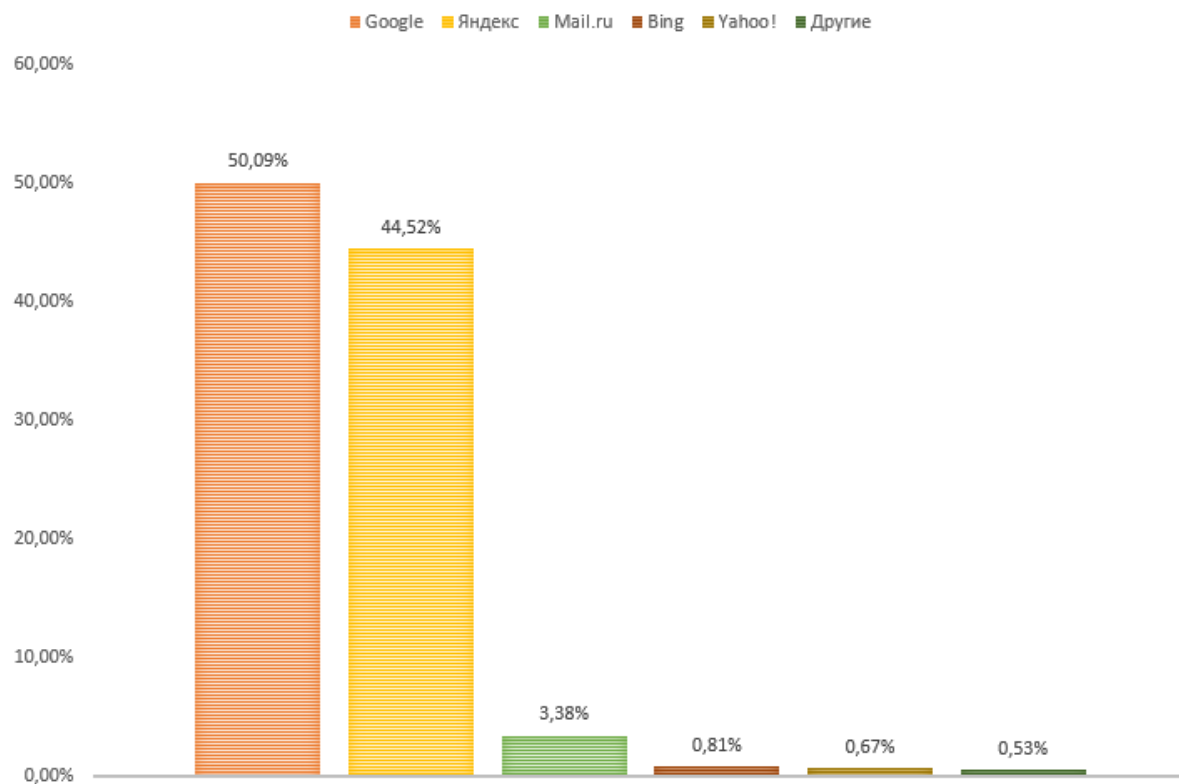
9. WolframAlpha

Отличительной чертой этого поиска является многообразие вспомогательных сервисов, рассчитанных на запросы, связанные с теми или иными знаниями. То есть, вы не увидите ссылок на посты в соцсетях или статьи желтой прессы. Вам предложат конкретные цифры и проверенные факты в форме единого документа. Этот браузер идеально подходит школьникам и студентам.

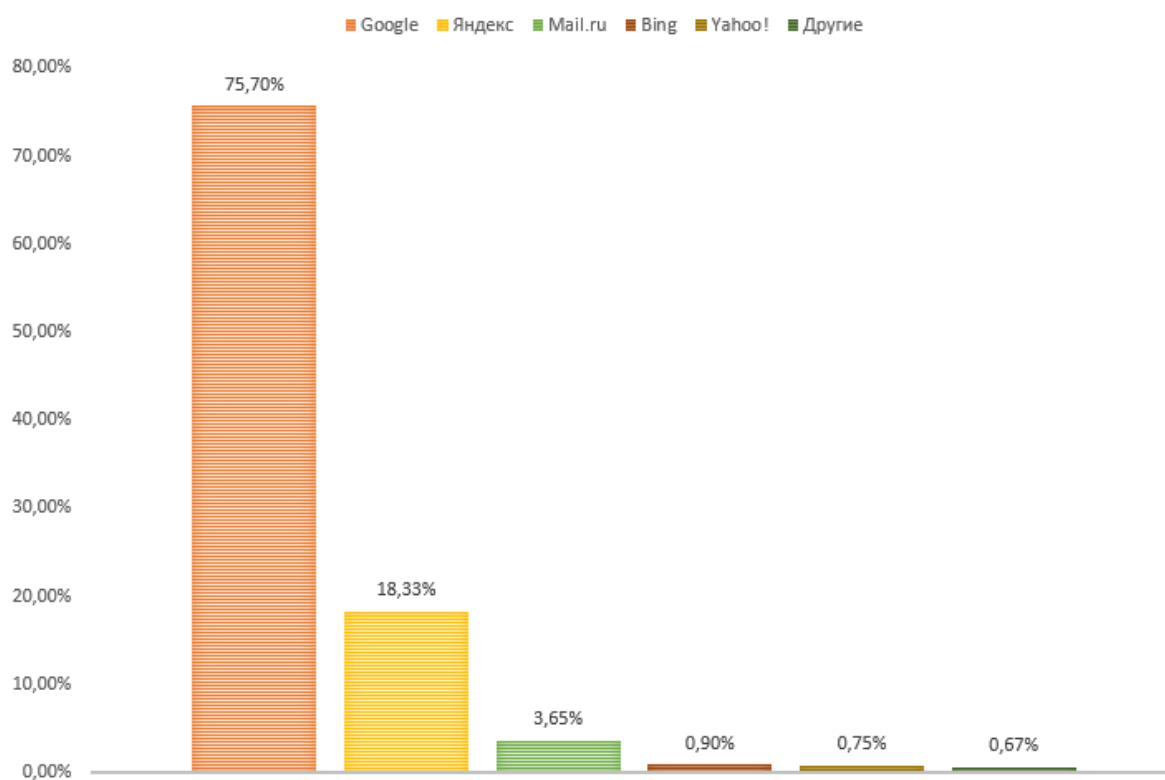
10. Yandex

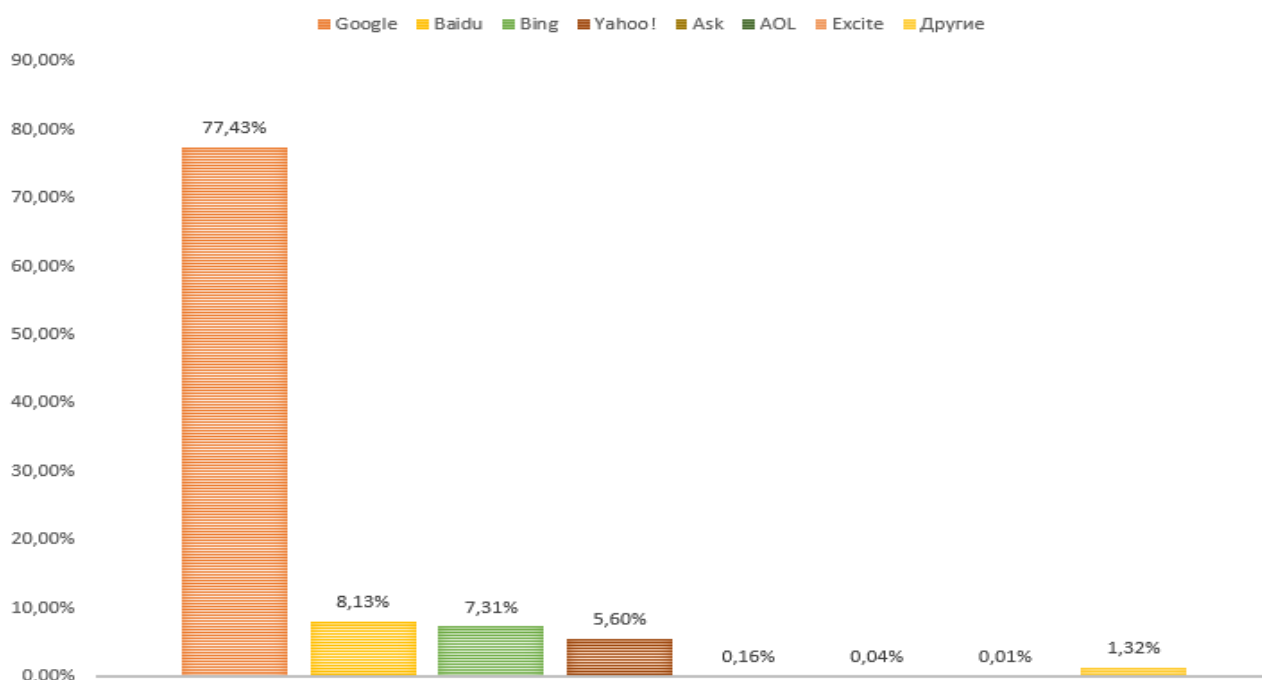
Поисковик, наиболее популярный в России и странах СНГ. Кроме того, около 3% аудитории сайта являются жителями Германии. Сайт примечателен большим количеством сервисов на все случаи жизни (музыка, радио, расписание общественного транспорта, недвижимость, переводчик и пр.) Также ресурс предлагает большой выбор индивидуального оформления сайта, а также настройку виджетов «под себя». Яндекс занимает 31 место в мире по популярности, потеряв 11 позиций за последний год.

РОССИЯ, МАЙ 2017



УКРАИНА, МАЙ 2017





10 Средства редактирования HTML-кода

Для создания любого файла нужна какая-нибудь программа.

1. Простые текстовые редакторы и текстовые процессоры (Word, Блокнот)
2. Текстовые редакторы с подсветкой тегов (Notepad++, PSPad)
3. Специализированные редакторы HTML-кода

Специализированные HTML-редакторы отличаются тем, что поддерживают следующие возможности:

- подсветка синтаксиса — выделение тегов, текста, ключевых слов и параметров разными цветами. Это облегчает поиск нужного элемента, ускоряет работу разработчика и снижает возникновение ошибок;
- работа с вкладками. Сайт представляет собой набор файлов, которые приходится править по отдельности, для чего нужен редактор, умеющий одновременно работать сразу с несколькими документами. При этом файлы удобно открывать в отдельных вкладках, чтобы быстро переходить к нужному документу;
- подсказки по использованию элементов html и css (встроенные справочники);
- проверка текущего документа на ошибки;
- встроенный браузер, что позволяет, не выходя из редактора посмотреть, как будет в итоге выглядеть создаваемая страничка.

Специализированные редакторы HTML-кода подразделяются на типы:

- редакторы тегов (HTMLPad, LightPad, HomeSite, Visual HTML Workshop) — предполагают, что пользователь знаком с языком HTML
- WYSIWYG-редакторы (Netscape Composer, FrontPage Express, Adobe (Macromedia) Dreamweaver) — часто называют редакторами типа «что видишь, то и получишь» (what you see is what you get).