

Лекция 1 Введение в веб-технологии

Веб-технологии. Понятие, особенности, применения. Сайты и страницы. Веб-сервер. Принципы построения взаимодействия клиента и сервера. Понятие о клиентских и серверных приложениях.

1.1 Термины и понятия

Определение 1. **Web, World Wide Web, Всемирная паутина, ППП (повсеместно протянутая паутина)** - глобальное информационное пространство, основанное на физической инфраструктуре Интернета и протоколе передачи данных HTTP.

Определение 2. **Internet, Интернет, «Инет», Internet, Interconnected Networks, «Всемирная сеть», «Глобальная сеть», «Всемирная паутина»** — всемирная система добровольно объединённых компьютерных сетей, построенная на использовании протокола IP и маршрутизации пакетов данных. Интернет образует всемирную (единую) информационную среду — способ организации оцифрованной информации. Интернет служит физической основой для Всемирной паутины.

В обывденной речи в настоящее время Web, Веб, Internet ,Интернет – почти синонимы и употребляются без разделения понятий.

Определение 3. **Веб-технологии (иногда также употребляется в единственном числе) – комплекс технических, коммуникационных, программных методов решения задач организации совместной деятельности пользователей с применением сети Интернет.**

Определение 4. **Контент** – информация, размещенная на веб-страницах.

Определение 5. **URI страницы** – уникальный адрес страницы в сети Интернет.
Пример: http://www.yandex.ru/all_services.html

URI включает:

1. Метод доступа к ресурсу, т. е. протокол доступа (http).
2. Сетевой адрес ресурса (www.yandex.ru)
3. Полный путь к файлу на сервере (all_services.html).

1.2 Web-технологии – что это такое?

1.2.1 Особенности

Web-технологии есть **концепция работы с информацией**. Она отличается следующими особенностями:

- техническая основа Web-технологий – локальные и глобальные сети, часто Интернет
- применение особого типа тонких клиентов: web-браузеров (типы и история, современное состояние браузеров отдаются на самостоятельное изучение)
- преимущественно текстовая и статично-графическая подача информации (ослабление этой тенденции связано с развитием технологий связи и ПО, экспансией медиаконтента)
- изменения в информационных источниках мгновенно отражаются в публикациях;
- число потребителей информации практически не ограничено. публикатор сам может задать особые условия на доступ к публикуемой информации;
- в публикациях могут содержаться ссылки на другие публикации без ограничения на местоположение и источники материалов;
- активная работа поисковых машин (история, современное состояние и роль поисковых машин отдаются на самостоятельное изучение);
- доставка и тиражирование контента практически бесплатны.

1.2.2 Достоинства

Привлекательность Web-технологий как средства доставки информации во многом определяет универсальный интерфейс между человеком и компьютером. Каждому человеку понятны надписи, заголовки, ссылки, картинки. Веб-интерфейс как средство доступа к информации интуитивно понятен.

Следствием простоты веб-интерфейса является широкая употребимость Интернета как канала коммуникации.

Браузер – программа для просмотра веб-страниц и работы с информацией в веб-интерфейсе. Браузеры – программы, которыми обеспечены все современные компьютеры большое число т.н. «гаджетов».

Теоретически все браузеры должны отображать все сайты, сделанные по стандартам, одинаково. Практически имеется множество тонкостей и сложностей. Наиболее популярные браузеры: Internet Explorer, FireFox, Opera, Safari, Chrome.

1.2.3 Интеграционная роль веб-технологий.

Значение Web-технологии, как для разработчиков программного обеспечения, так и для обычных пользователей во многом определяется тем, что это, прежде всего – **интеграционная технология**. И трудно найти более удачный пример того, как можно интегрировать различные источники информации и различные ее типы.

Веб-технологии позволяют создавать простые для освоения, легкодоступные, крайне дешевые, быстро обновляемые информационные, диалоговые, справочные системы.

1.3 Сайты и страницы, сервисы, порталы

Определение 6. Web-страница (гипертекстовый документ) — это документ, описанный на языке HTML. Основное отличие их от текстовых документов

состоит в том, что они могут включать ссылки на другие аналогичные документы.

Страницей называют то, что показывает браузер при вводе адреса страницы или при переходе по ссылке.

Определение 7. **Сайтом (веб-сайтом)** называют совокупность страниц, созданных с применением программного обеспечения и образующая единое целое в техническом, информационном и навигационном аспектах. Чаще всего все страницы сайта имеют общее доменное имя (понятие доменного имени, системы DNS-серверов, структура www отдаются на самостоятельное изучение).

Определение 8. **Сервис (веб-сервис)** – специализированный сайт для решения нужных посетителям достаточно узких задач. Пример сервиса – gmail.com или www.whatthefont.com. У слова есть и другое значение (см <http://ru.wikipedia.org/wiki/Web-сервис/>)

Определение 9. **Web-сервер (более точно — http-сервер)** — специальная программа, установленная на общедоступном компьютере. Web-сервер обеспечивает одновременную обработку запросов от множества клиентов по протоколу HTTP.

Определение 10. **Портал**. Выделяют два значения. 1. Крупный тематический сайт, активно развиваемый и посещаемый, с четко определенной и растущей аудиторией и постепенно пополняемый сервисами. Примеры: free-lance.ru, www.RussianRealty.ru 2. Сайт без выраженной тематической направленности или с широкой тематикой (например, географической), имеющий в своем составе множество сервисов, служб и направлений взаимодействия. Примеры: yandex.ru, volga34.ru

1.4 Веб-сервер и браузер. Взаимодействие.

Все сайты открываются браузерами как html-документы. html-документ может содержать изображения, видеофрагменты, flash-ролики и звуки.

Все современные сайты имеют динамические элементы, то есть фрагменты контента, которые меняются во времени, а так же в зависимости от того, кто именно зашел на страницу, и даже могут редактироваться самими посетителями.

Для того, чтобы эти функциональные элементы работали, необходимо, чтобы сервер, выдающий страницы, имел специализированное программное обеспечение, работающее с базой данных и создающее страницы динамически. (Представление о том, что все страницы всех сайтов, вплоть до выдачи поисковых машин, формируются в html вручную, является примитивным, но крайне распространенным заблуждением.)

Браузер и веб-сервер взаимодействуют по технологии клиент-сервер. После ввода адреса в строку адреса браузер формирует запрос к серверу. Сервер формирует страницу и передает ее браузеру. Браузер выводит страницу пользователю, который своими действиями формирует новый запрос.



Для того, чтобы HTML-страница стала динамической, то есть могла зависеть от поведения человека и/или внешних событий, существует несколько технологий. Первое деление связано с местом осуществления этого оживления: на сервере или на клиенте. Далее деление ведется по методу программирования: с использованием интерпретируемых скриптов или откомпилированных программ.

Страна клиента		Страна сервера	
Скрипты	Программы	Скрипты	Программы
JavaScript	Plug-ins	Perl	отдельные CGI-программы
JScript	ActiveX	Tcl	модули сервера
VBScript	Java applets	Shell-код	ISAPI/NSAPI
			ActiveX
			Java servlets

1.5 Веб-браузеры

- Современный браузер выполняет следующие функции:
- обеспечение передачи данных по протоколам http и https;
 - обработка контента веб-страницы (css, javascript, медиа-файлы);

Практически все популярные браузеры распространяются бесплатно или «в комплекте» с другим приложением. Браузеры классифицируются:

- 1) По-типу устройства для которого написаны:
 - браузеры ([Internet Explorer](#), [Firefox](#), [Safari](#), [Opera](#));
 - браузеры для портативных устройств ([Internet Explorer Mobile](#), [Opera Mini](#), [Safari](#) для Apple [iPhone](#)).
- 2) По-типу интерфейса:
 - графические браузеры (все выше названные);
 - текстовые браузеры ([Lynx](#), [Links](#), [W3M](#), [Netrik](#), [Elinks](#), [Internet Browser](#)).
- 3) По-режиму доступа:
 - online-браузеры (все выше перечисленные) - браузер пытается получить страницы с веб-сервера;
 - offline-браузеры (HTTrack, Offline Explorer, Teleport Pro, WebZip) – просматривать можно только сохранённые копии веб-сайтов или их частей.

Современное состояние веб-браузеров

При посещении веб-сайта клиентское приложение (такое как [браузеры](#), [поисковые роботы](#)) обычно посылает [веб-серверу](#) информацию о себе. Это текстовая строка, являющаяся частью [HTTP](#) запроса, начинающаяся с **User-Agent:**, и обычно включающая такую информацию, как название и версию приложения, операционную систему компьютера и язык. У «поисковых роботов» эта строка часто содержит [URL](#) и [email](#) адрес, по которым веб-мастер может связаться с оператором «робота».

По данным компании [Net Applications](#) в мае 2008 года рыночная доля браузеров со строкой [User Agent](#)

Таблица – статистика по User Agent

Microsoft Internet Explorer	73.75%
Firefox	18.41%
Safari	6.25%
Opera	0.71%
Netscape	0.62%
Mozilla	0.08%
Opera Mini	0.05%
Playstation	0.03%
Microsoft Pocket Internet Explorer	0.02%
Blazer	0.02%
Konqueror	0.02%
ACCESS NetFront	0.01%
WebTV	0.01%
Lotus Notes	0.00%
ANT Galio	0.00%
BlackBerry	0.00%
Danger Web Browser	0.00%
iCab	0.00%

1.6 Протокол HTTP

Определение 11.**HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста»)** — протокол прикладного уровня передачи данных.

HTTP в настоящее время повсеместно используется для получения информации с веб-сайтов.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (англ. Uniform Resource Identifier) в запросе клиента.

HTTP — протокол прикладного уровня (см уровни модели OSI). Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ».

В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ».

Компоненты, использующие HTTP (браузер и сервер), могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами. Браузер, посылающий запросы, может отслеживать задержки ответов. Сервер может хранить IP-адреса и заголовки запросов последних клиентов. Однако сам протокол не осведомлён о предыдущих запросах и ответах, в нём не предусмотрена внутренняя поддержка состояния, к нему не предъявляются такие требования.

1.6.1 Структура протокола HTTP

Каждое HTTP-сообщение состоит из трёх частей, которые передаются в указанном порядке:

1. Стартовая строка (англ. Starting line) — определяет тип сообщения;
2. Заголовки (англ. Headers) — характеризуют тело сообщения, параметры передачи и прочие сведения;
3. Тело сообщения (англ. Message Body) — непосредственно данные сообщения. Обязательно должно отделять от заголовков пустой строкой.

Заголовки и тело сообщения могут отсутствовать, но стартовая строка является обязательным элементом, так как указывает на тип запроса/ответа.

Стартовые строки различаются для запроса и ответа. Строка запроса выглядит так:

GET URI — для версии протокола 0.9.

Метод URI HTTP/Версия — для остальных версий.

Здесь:

- Метод (англ. Method) — название запроса, одно слово заглавными буквами. В версии HTTP 0.9 использовался только метод GET, список запросов для версии 1.1 представлен ниже.
- URI определяет путь к запрашиваемому документу.

Стартовая строка ответа сервера имеет следующий формат:
HTTP/Версия КодСостояния Пояснение

Здесь:

- Версия — пара разделённых точкой арабских цифр как в запросе.
- КодСостояния (англ. Status Code) — три арабские цифры. По коду статуса определяется дальнейшее содержимое сообщения и поведение клиента.
- Пояснение (англ. Reason Phrase) — текстовое короткое пояснение к коду ответа для пользователя. Никак не влияет на сообщение и является необязательным.

В случае успешного выполнения запроса сервер ответит строкой: HTTP/1.0 200
Ok

1.6.2 Методы запросов протокола HTTP

Определение 12. **Метод HTTP (англ. HTTP Method)** — последовательность из любых символов кроме управляющих и разделителей, указывающая на основную операцию над ресурсом. Обычно метод представляет собой короткое английское слово записанное заглавными буквами.

Название метода чувствительно к регистру. Если метод серверу неизвестен, от отвечает ошибкой 501 (Method not implemented). Если серверу метод известен, но он не применим к конкретному ресурсу, то возвращается сообщение с кодом 405 (Method Not Allowed).

Наиболее часто используемые методы это GET, HEAD и POST.

GET. Используется для запроса содержимого указанного ресурса.

Клиент может передавать параметры выполнения запроса в URI целевого ресурса после символа «?»:

GET /path/resource?param1=value1¶m2=value2 HTTP/1.1

Согласно стандарту HTTP, многократное повторение одного и того же запроса GET должно приводить к одинаковым результатам. Это позволяет кэшировать ответы на запросы GET.

HEAD. Аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело. Запрос HEAD обычно применяется для извлечения метаданных, проверки наличия ресурса (валидация URL) и чтобы узнать не изменился ли он с момента последнего обращения.

POST. Применяется для передачи пользовательские данные заданному ресурсу. Например, в блогах посетители обычно могут вводить свои комментарии к записям в HTML-форму, после чего они передаются серверу методом POST и он помещает их на страницу. При этом передаваемые данные (в примере с блогами — текст комментария) включаются в тело запроса. Аналогично с помощью метода POST обычно загружаются файлы.

В отличие от метода GET, для метода POST многократное повторение одних и тех же запросов POST может возвращать разные результаты (например, после каждой отправки комментария будет появляться одна копия этого комментария). Сообщение ответа сервера на выполнение метода POST не кэшируется.

Существуют также методы: PUT, PATCH, DELETE, TRACE, CONNECT, LINK, UNLINK

1.6.3 Коды состояния протокола HTTP

Определение 13. **Код состояния протокола HTTP** – числовое значение состояния протокола, определяемое результатом обработки последнего запроса.

В настоящее время выделено пять классов кодов состояния.

1xx Informational (Информационный)

2xx Success (русск. Успешно)

Сообщения данного класса информируют о случаях успешного принятия и обработки запроса клиента. В зависимости от статуса сервер может ещё передать заголовки и тело сообщения.

3xx Redirection (русск. Перенаправление)

Коды статуса класса 3xx сообщают клиенту что для успешного выполнения операции нужно произвести следующий запрос к другому URI. В большинстве случаев новый адрес указывается в поле Location заголовка. Клиент в этом случае должен, как правило, произвести автоматический переход.

4xx Client Error (русск. Ошибка клиента)

Класс кодов 4xx предназначен для указания ошибок со стороны клиента. При использовании всех методов кроме HEAD сервер должен вернуть в теле сообщения гипертекстовое пояснение для пользователя.

5xx Server Error (русск. Ошибка сервера)

Коды 5xx выделены под случаи неудачного выполнения операции по вине сервера. Для всех ситуаций, кроме использования метода HEAD, сервер должен включать в тело сообщения объяснение, которое клиент отобразит пользователю.

1.6.4 Пример диалога по протоколу HTTP

Запрос:

```
GET /wiki/HTTP HTTP/1.1
```

```
Host: ru.wikipedia.org
```

```
User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509  
Firefox/3.0b5
```

```
Accept: text/html
```

```
Connection: close
```

Ответ:

```
HTTP/1.0 200 OK
```

```
Server: nginx/0.6.31
```

```
Content-Language: ru
```

```
Content-Type: text/html; charset=utf-8
```

```
Content-Length: 1234
```

```
Connection: close
```

```
<содержимое запрошенной страницы>
```

1.7 Клиентские скрипты

Как следует из названия, клиентские языки обрабатываются на стороне клиента пользователя, а если проще - программы на клиентском языке обрабатывает браузер.

Отсюда следует и недостаток – это то, что обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так,

чтобы он вообще игнорировал написанные вами скрипты. При этом, если браузер старый, он может не поддерживать тот или иной язык или версию языка.

Также код клиентского скрипта может посмотреть каждый, кто откроет страницу со скриптом.

Преимущество же клиентского языка заключается в том, что обработка скриптов на таком языке может выполняться без отправки документа на сервер.

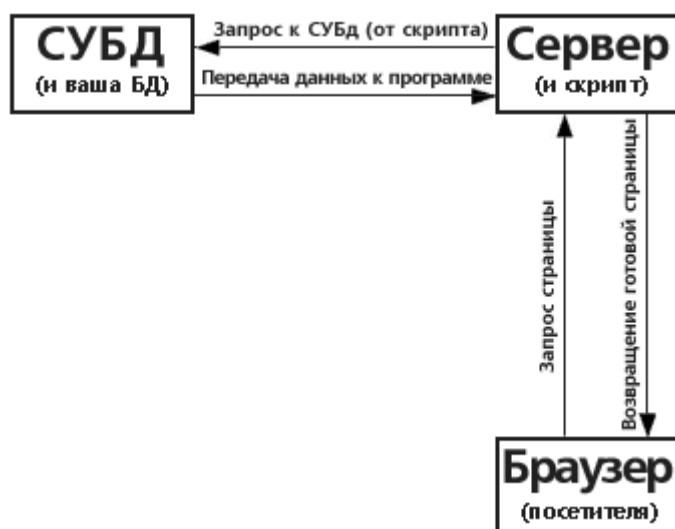
Пример: нужно проверить правильно ли пользователь ввел e-mail; чтобы это сделать пользователю, надо было бы отправить форму с заполненными данными, потом дожидаться, пока она обработается, и лишь после этого получить сообщение об ошибке (если она, разумеется, присутствует). Это недопустимо с точки зрения удобства для пользователя и траты ресурсов.

С клиентским же языком 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, то есть, например, с его помощью нельзя сделать гостевую книгу.

Самым распространенным из языков клиентских скриптов является JavaScript.

1.8 Серверные скрипты

Обработка скриптов на серверном языке.



Когда пользователь дает 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 скрипты, связанные со страницей, и только потом возвращается к посетителю в виде простого HTML-документа (то есть посетитель уже никак не сможет увидеть код серверного скрипта).

Работа серверных скриптов зависит от сервера, на котором расположен сайт, и от того, какие технологии поддерживаются сервером.

Серверные языки программирования открывают перед программистом большие функциональные возможности. Современные сайты зачастую представляют собой

чрезвычайно сложные программно-информационные системы, решающие значительное количество задач бизнес-логики и теоретически могут дублировать функции большинства бизнес-приложений.

Серверные скрипты, как правило, взаимодействуют с базами данных. Наиболее часто используются СУБД MySQL, PostgreSQL, MS SQL Server, Oracle.

Среди языков серверных скриптов наиболее распространенными являются PHP, perl, Asp.net.

1.9 Литература

1. [Уроки веб-мастерства \(http://www.igp.ru/sit/Mariya/HTML-1-02-84.htm\)](http://www.igp.ru/sit/Mariya/HTML-1-02-84.htm)
2. <http://ito.osu.ru/resour/fpk/TRiP/lecture/texthtml/slovar.htm>
3. <http://ru.wikipedia.org/wiki/Http>
4. <http://ru.wikipedia.org/wiki/www>
5. <http://ru.wikipedia.org/wiki/веб-сервер>
6. [Первые шаги - Языки веб-программирования: клиентские и серверные \(www.postroika.ru/programming/webpro.html \)](http://www.postroika.ru/programming/webpro.html)