

# Адресация в Интернет

Любой компьютер, подключающийся к Интернет, должен иметь уникальный IP-адрес. До недавнего времени любой адрес представлялся в виде 4 байт и записывался четырьмя целыми числами, разделенными точками, например, 112.15.38.127. Такая система адресации, называемая **IPv4**, имеет ограничение: количество возможных адресов порядка 4,2 млрд.

В последние годы активно внедряется система IPv6, в которой на адрес отводится 16 байт. Количество возможных адресов в этом случае таково, что можно присвоить свой адрес каждой молекуле в земной атмосфере. Адреса в **IPv6** записываются в шестнадцатеричной системе, содержимое каждых двух байт отделяется двоеточием: fe80:0:0:0:200:f8ff:fe21:67cf. В настоящее время в Интернете поддерживаются обе системы адресации.

# Адресация в Интернет

**DNS (Domain Name System** — система доменных имён) — это иерархическая система различных групп компьютеров, имеющих свои имена.

Пример DNS-имени компьютера:



имя сервера      поддомен (домен 2-го уровня)      домен 1-го уровня

Различают международные домены первого уровня (букв в имени больше двух) и национальные (двухбуквенные).

# Адресация в Интернет

Домены по виду деятельности. В 1985 году было предложено 7 доменов:

- .com — для коммерческих сайтов;
- .edu — для образовательных сайтов;
- .gov — для сайтов государственных организаций США;
- .mil — для военных организаций США;
- .net — для сайтов, чья деятельность связана с Сетью;
- .org — для некоммерческих организаций;
- .int — для международных организаций.

Сейчас этот список существенно расширен.

# Адресация в Интернет

Национальные домены отражают принадлежность сайта определенной стране, например:

- fi – Финляндия
- de – Германия
- ru – Россия
- tv – Тувалу
- uk – Великобритания.

Сайт может иметь несколько DNS-адресов, но только один IP-адрес.

# Адресация в Интернет

Поскольку для установления связи и передачи данных сетевые протоколы используют IP-адреса, в сети имеются **DNS-серверы** – это компьютерные программы и компьютеры, задачей которых является **определение IP-адреса по DNS-адресу**.

DNS-серверы связаны между собой в иерархическую структуру. Передача запроса на поиск соответствия может производиться как по горизонтали, так и по вертикали (в сети другого уровня). DNS-сервер более высокого уровня периодически копирует таблицы подчиненных DNS-серверов. Если соответствие между DNS и IP-адресами не было найдено на одном, будет осуществлен поиск на соседних DNS-серверах. Каждый DNS-сервер, не нашедший соответствия, инициирует запрос к своим соседям, в результате за относительно короткое время либо будет найден соответствующий IP-адрес, либо будет выдано сообщение о его отсутствии.

# Службы Интернет

Основными Интернет-службами (сервисами) являются:

- WWW (World Wide Web) – работа с информацией в виде гипертекстовых страниц (Web-страниц);
- FTP (File Transfer Protocol) – передача файлов;
- E-mail – электронная почта;
- UseNet (News) – телеконференции (группы новостей);
- IRC (Chat) – чат, обмен мгновенными сообщениями;
- Потокковое мультимедиа – передача звуковой и видеоинформации.

# Службы Интернет

Традиционную технологию создания Web-страниц, когда информация на страницах размещается теми, кто ее создавал, называют **Web 1.0**.

Технология **Web 2.0** предполагает активное участие самих пользователей в наполнении страниц информацией. Страница становится тем лучше, чем больше людей ею пользуются.

Примеры использования технологии Web 2.0:

- социальные сети,
- интернет-энциклопедии (Wikipedia, Wikimapia, Wikionary и др.),
- хранилища видеозаписей (YouTube),
- блоги (Twitter).

# Облачные сервисы

Облачные вычисления заключаются в том, что пользователь в любой момент может через Интернет получить доступ к нужной программе или данным, в связи с чем нет необходимости все это хранить в своем компьютере.

Национальным институтом стандартов и технологий США зафиксированы следующие обязательные характеристики *облачных вычислений*:

- *Самообслуживание по требованию* (англ. self service on demand) - потребитель самостоятельно определяет и изменяет вычислительные потребности, такие как серверное время, скорости доступа и обработки данных, объём хранимых данных без взаимодействия с представителем поставщика услуг;
- *Универсальный доступ по сети* - услуги доступны потребителям по сети передачи данных вне зависимости от используемого терминального устройства;



# Облачные сервисы

- *Объединение ресурсов* (англ. resource pooling) - поставщик услуг объединяет ресурсы для обслуживания большого количества потребителей в единый пул для динамического перераспределения мощностей между потребителями в условиях постоянного изменения спроса на мощности. Фактическое распределение ресурсов, предоставляемых потребителю, осуществляет поставщик.
- *Эластичность* - услуги могут быть предоставлены, расширены, сужены в любой момент времени, без дополнительных издержек на взаимодействие с поставщиком, как правило, в автоматическом режиме.
- *Учёт потребления* - поставщик услуг автоматически исчисляет потреблённые ресурсы (например, объём хранимых данных, размер трафика, количество пользователей, количество транзакций), и на основе этих данных оценивает объём предоставленных потребителям

# Облачные сервисы

## Достоинства:

- С точки зрения поставщика, благодаря объединению ресурсов и непостоянному характеру потребления со стороны потребителей, облачные вычисления позволяют экономить на масштабах, используя меньшие аппаратные ресурсы, чем требовались бы при выделенных аппаратных мощностях для каждого потребителя, а за счёт автоматизации процедур модификации выделения ресурсов существенно снижаются затраты на абонентское обслуживание.
- С точки зрения потребителя, возможность получить услуги с высоким уровнем доступности и низкими рисками неработоспособности, обеспечить быстрое масштабирование вычислительной системы благодаря эластичности без необходимости создания, обслуживания и модернизации собственной аппаратной

инфраструктуры

# Облачные сервисы

## Недостатки:

- Проблема создания неконтролируемых данных, когда информация, оставленная пользователем, будет храниться годами, либо без его ведома, либо он будет не в состоянии изменить какую-то её часть.
- Возможная потеря данных или их утечка.
- Деятельность организации зависит от надежности и пропускной способности интернет-соединения.

# Языковые средства создания сайтов

**HTML (HyperText Markup Language)** - язык разметки гипертекста. Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

HTML-документ – это обычный текстовый документ, созданный в любом текстовом редакторе, например в Блокноте, и оформленный в соответствии с правилами языка HTML.

Для файлов, содержащих HTML-документы, используется расширение HTML или HTM.

Гипертекстовый документ состоит из элементов.

**Элемент** – это конструкция языка HTML, содержащая данные. HTML включает в себя различные типы элементов, которые позволяют задавать абзацы, гипертекстовые ссылки, списки, таблицы, изображения и т. д.

# Языковые средства создания сайтов

Пример элемента:

```
<P>Привет!</P>
```

<P> –открывающий тег.

Привет! - содержание элемента,

</P> закрывающий тег.

Название элемента (P) присутствует и в открывающем, и в закрывающем теге. Регистр символов в названии элемента не имеет значения. Открывающий и закрывающий теги нужны для указания начала и конца элемента.

Теги всегда начинаются символом < и заканчиваются символом >. В закрывающем теге перед его именем помещается символ /. Для некоторых типов элементов допускается отсутствие закрывающего тега (например, элемент P, указывающий начало абзаца). Существуют также элементы, не имеющие закрывающего тега.

# Языковые средства создания сайтов

Элементы могут содержать параметры, называемые **атрибутами**. Атрибуты могут иметь определенные значения (по умолчанию или устанавливаемые авторами). Пара атрибут/значение указывается в **начальном теге** элемента перед символом `>`, например:

```
<BODY bgcolor = «#FF0000»>.
```

Каждому атрибуту может быть присвоено значение определенного типа. В приведенном примере указание атрибута `bgcolor` в элементе `BODY` приведет к тому, что цвет фона страницы станет красным.

Значения атрибутов заключаются в кавычки, регистр, в котором записываются названия атрибутов, не имеет значения.

В начальном теге элемента может быть указано любое количество допустимых пар атрибут/значение, разделенных пробелами, причём, порядок записи атрибутов не имеет

# Языковые средства создания сайтов

Элемент, находящийся внутри другого элемента, называется **вложенным**.

Пример использования вложенных элементов для задания начертания шрифта:

`<I>Курсив<B> Полужирный курсив </B>Курсив</I>`.

При использовании вложенности следует помнить, что вложенные элементы должны закрываться до того, как будут закрыты внешние элементы.

Пример **неверной** вложенности элементов:

`<I>Неправильное закрытие<B> внутреннего элемента</I>  
до закрытия внешнего</B>`

Если по каким-то причинам (например, при ошибке в записи тега) тег не распознается браузером, то он игнорируется, никакие **сообщения об ошибках не выдаются**.

# Языковые средства создания сайтов

Многие виды ресурсов, размещенных в Интернете чаще всего представляют собой файлы на жестком диске компьютера (сервера), подключенного к сети.

С каждым ресурсом сопоставляется запись, по которой можно однозначно определить его расположение, – **универсальный идентификатор ресурса** или **URI (Universal Resource Identifier)**.

URI состоит из трех частей:

- наименование протокола передачи данных,
- доменное имя компьютера
- путь к файлу ресурса.

`http://www.site.ru/info/examples/ex_1.html`

←  
протокол

←  
доменное имя

←  
путь к файлу



# Языковые средства создания сайтов

При помощи URI можно также ссылаться **на части HTML-документов**, например:

`http://www.somesite.com/info/examples/ex_1.html#description`

При использовании этого URI можно получить доступ к части HTML-документа, имеющей имя `description`.

URI также позволяют ссылаться **на ресурсы в пределах одного компьютера**. При этом указывается относительный путь ресурса. Например, чтобы из HTML-документа, расположенного в папке `/info/examples`, сослаться на файл `/info/files/file1.jpg`, достаточно задать URI `/files/file1.jpg`.

В HTML-документах при помощи подобных ссылок указываются пути рисунков и других объектов, используемых в документах, но непосредственно не хранимых в них.

# Языковые средства создания сайтов

В идеальном случае HTML-документ состоит из трех частей:

- данные о версии используемого HTML;
- заголовок документа;
- тело документа.

```
<!DOCTYPE HTML PUBLIC « //W3C//DTD HTML 4.01//EN»>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Мой первый HTML документ</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Заголовок</H1>
```

```
<P>Первый абзац
```

```
<P>Второй абзац
```

```
</BODY>
```

```
</HTML>
```

# Языковые средства создания сайтов

Корневым элементом структуры HTML-документа является элемент HTML. Его использование позволяет явно указать браузеру, что им обрабатывается HTML-код. Элемент HTML содержит в себе все остальные структурные части HTML-документа.

Заголовок HTML-документа содержится в элементе HEAD. Он не является обязательным.

В самом простом случае в заголовке документа содержится только один элемент – TITLE. Он используется для задания названия HTML-документа. Этот элемент задается при помощи парных тегов <TITLE> и </TITLE>, причем только один раз. Текст, находящийся между приведенными тегами, воспринимается и отображается браузерами (и поисковыми системами) как название документа. К тексту названия форматирование неприменимо.

# Языковые средства создания сайтов

Вся содержательная часть HTML-документа находится в его теле (элемент BODY). Для определения этого элемента используются парные теги <BODY> и </BODY>. Теги <BODY> и </BODY> не являются обязательными, но их наличие, как и в случае тегов <HEAD> и </HEAD>, значительно улучшает наглядность структурной организации HTML-документа и позволяет четко отделить содержимое документа от заголовка.

В текст HTML-документа могут вставляться элементы, содержащие программы на других языках (VbScript, JavaScript и др.).

VbScript, JavaScript – это интерпретируемые языки программирования, которые позволяют выполнять программу (сценарий) на стороне клиента. За счет этого можно повысить функциональность сайта, добавить ему такие возможности, которые не предоставляет HTML.