



# Лекция 2

## Клиентские и серверные веб-технологии

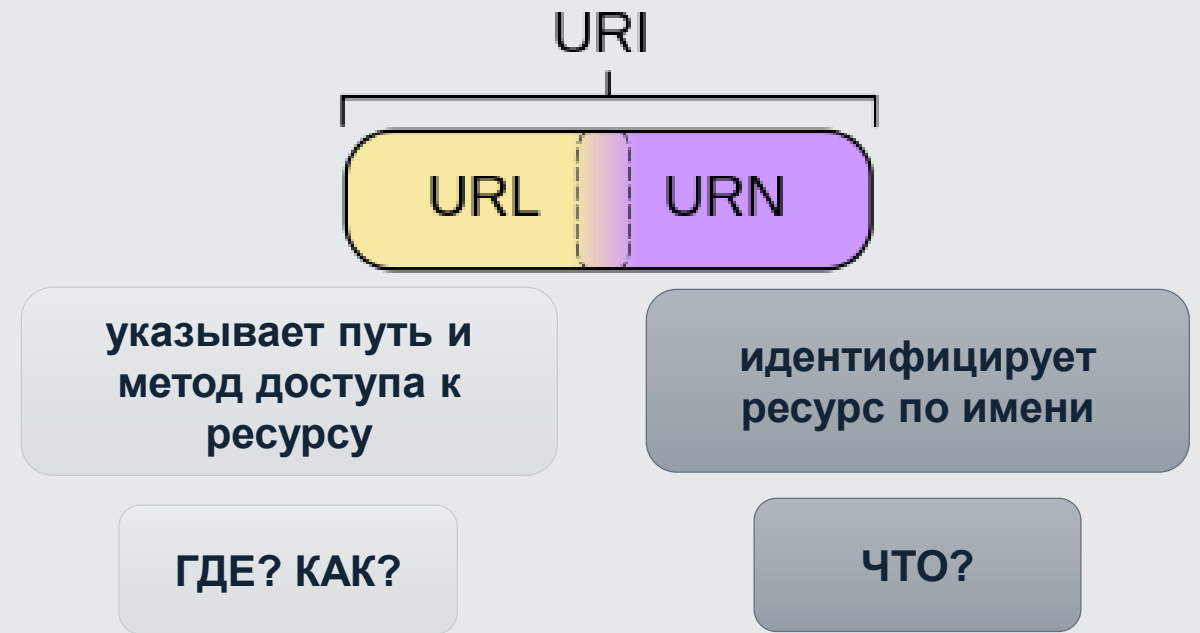
Introduction to Web Development

# План лекции

- Понятия клиентских и серверных веб-технологий
- Обзор клиентских технологий
- Обзор серверных технологий

# Понятия URI, URL, URN

URI (Uniform Resource Identifier) — унифицированный (единообразный) идентификатор ресурса.



URI — это последовательность символов, идентифицирующая абстрактный или физический ресурс.

**URI** может указывать:

- местоположение ресурса (URL - Uniform Resource Locator),
- его имя (URN - Uniform Resource Name),
- и то и другое.

# Понятия URI, URL, URN

## Структура URI:

[ схема ":" ] иер-часть [ "?" запрос ] [ "#" фрагмент ]

### Схема

- **схема обращения к ресурсу** (часто - сетевой протокол), например *http*, *ftp*, *file*, *ldap*, *mailto*, *urn*

### Иер-часть

- обычно **путь к ресурсу** (и, возможно, перед ним, адрес сервера, на котором тот располагается) **или идентификатор ресурса**

### Запрос

- необязательный компонент, служат для идентификации ресурса в пределах видимости URI-схемы

### Фрагмент

- необязательный компонент

### Абсолютные URI:

<http://ru.wikipedia.org/wiki/URI>  
<file:\\C:/file.wsd>  
<mailto:John.Doe@example.com>

### Ссылки URI:

</relative/URI/with/absolute/path/to/resource.txt>  
<../../resource.txt>  
<resource.txt>

# Понятия URI, URL, URN

## Структура

### URL:

<схема>://<логин>:<пароль>@<хост>:<порт>/<URL-путь>?<параметры>#<якорь>

схема

- схема обращения к ресурсу (часто - сетевой протокол)

логин

- имя пользователя, используемое для доступа к ресурсу

пароль

- пароль указанного пользователя

хост

- полностью прописанное **доменное имя хоста** в системе DNS  
или **IP-адрес хоста**

порт

- **порт хоста** для подключения

URL-путь

- уточняющая **информация о месте нахождения ресурса**; зависит от протокола.  
?параметр\_1=значение\_1&параметр\_2=значение\_2&параметр3=значение\_3

параметры

- **строка запроса** с передаваемыми (методом GET) **параметрами**.

якорь

- идентификатор «якоря», ссылающегося на некоторую часть (раздел) открываемого документа

<http://ru.wikipedia.org/w/index.php?title=TCP&veaction=edit&section=22>

# Понятия URI, URL, URN

## Структура URN:

**<URN> ::= "urn:" <NID> ":" <NSS>**

**<NID>**

**идентификатор пространства имён**; представляет собой синтаксическую интерпретацию NSS, не чувствителен к регистру.

**<NSS>**

**строка из определённого пространства**;

если в этой строке содержатся символы не из набора ASCII, то они должны быть закодированы в Юникоде (UTF-8) и предварены (каждый из них) знаком процента «%».

**urn:ed2k:354B15E68FB8F36D7CD88FF94116CDC1**

## Применение в magnet-ссылках:

**magnet:?xl=[Размер в байтах]&dn=  
[Имя файла (URL encoded)]&xt=urn:tree:tiger:[ ТТН хеш файла (Base32) ]**

# Понятия клиентских и серверных веб-технологий

**Клиентские технологии**  
(browser execution):  
выполняются в браузере

- **HTML / CSS**
- **Adobe Flash**
- **JavaScript**
- **XML**
- **JSON**
- **Java**

**Серверные технологии**  
(server execution):  
обрабатывает сервер и  
возвращает клиенту только HTML-код.

- **SQL**
- **PHP**
- **ASP**
- **VBScript**
- **Java**

# Понятия клиентских и серверных веб-технологий

На стороне сервера используются технологии и любые языки программирования, способные осуществлять вывод в стандартную консоль.

| Название | Лицензия      | Веб-сервер                       |
|----------|---------------|----------------------------------|
| ASP      | проприетарная | специализированный               |
| ASP.NET  | проприетарная | специализированный               |
| Java     | свободная     | множество, в том числе свободных |
| Perl     | свободная     | практически любой                |
| PHP      | свободная     | практически любой                |
| Python   | свободная     | практически любой                |
| Ruby     | свободная     | практически любой                |



# Понятия клиентских и серверных веб-технологий

**На стороне клиента используются:**

**Для реализации GUI**  
(Graphical user interface –  
графический интерфейс  
пользователя):

**HTML,  
XHTML**

**CSS**

**Для формирования и  
обработки запросов,  
создания интерактивного  
интерфейса:**

**ActiveX**

**Adobe Flash,  
Adobe Flex**

**JavaScript**

**Java**

**Silverlight**

# Обзор клиентских технологий: HTML



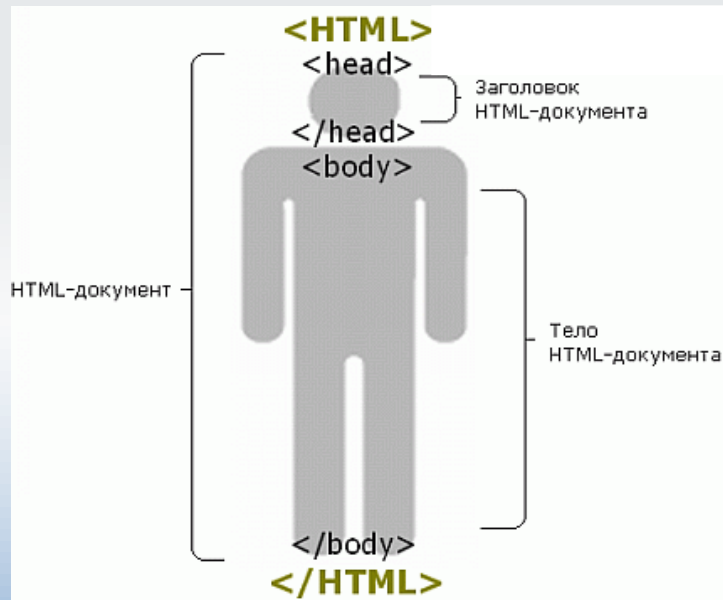
## HTML (HyperText Markup Language) – язык разметки гипертекста

Стандартный язык разметки документов.

Теговый язык.

```
<!DOCTYPE html>
<html>
  <!-- created 2010-01-01 -->
  <head>
    <title>sample</title>
  </head>
  <body>
    <p>Voluptatem accusantium
      totam rem aperiam.</p>
  </body>
</html>
```

HTML



Система верстки веб-страниц, которая определяет, какие элементы и как должны располагаться в документе.

HTML интерпретируется браузерами и отображается в виде документа в понятной для человека форме.

HTML-страница — это обычный текстовый файл, который содержит набор инструкций для интерпретации внутри браузеров.

# Обзор клиентских технологий: CSS

## CSS (Cascading Style Sheets) – каскадные таблицы стилей



```
body {  
  margin:2;  
  padding:5;  
  border:1;  
  width:100%;  
  background:#ffffff;  
  min-width:900px;  
  font-size:80%;  
}  
  
#header {  
  border-bottom:3px solid #aaaaaa;  
}  
  
#dimensions {  
  clear:both;  
  background:#eeeeee;  
  border-top:4px solid #ffffff;  
  border-bottom:4px solid #ffffff;  
  margin:4;  
  padding:6px 10px !important;  
  text-align:left;  
}
```

**Формальный язык описания внешнего вида документа**, написанного с использованием языка разметки.

Каскадные таблицы стилей дают возможность **хранить содержимое отдельно от его представления**.

Стандарт на основе текстового формата, **определяющий представление данных в браузере**.

Набор параметров форматирования, который применяется к элементам документа, чтобы **изменить их внешний вид**.

# Обзор клиентских технологий: CSS

## Стили CSS

### Связанные

Описание стилей **располагается в отдельном файле**, как правило, с расширением **css**;  
**для связывания документа с этим файлом применяется тег <link>**. Этот тег помещается в контейнер **<head>**.

### Глобальные стили

**CSS описываются в самом документе и располагаются в заголовке веб-страницы с помощью контейнера <style>**.

### Внутренние стили

Являются расширением для **одиночного тега** используемого на текущей веб-странице. Для определения стиля используется атрибут **style**, а его значение **выступает набор стилевых правил**.

# Обзор клиентских технологий: CSS

## Связанные стили

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Стили</title>
    <link rel="stylesheet" type="text/css" href="mysite.css">
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
    <h1>Заголовок</h1>
    <p>Текст</p>
  </body>
</html>
```

### Содержимое файла mysite.css

```
H1 {
  color: #000080;
  font-size: 200%;
  font-family: Arial, Verdana, sans-serif;
  text-align: center;
}
P {
  padding-left: 20px;
}
```

# Обзор клиентских технологий: CSS

## Глобальные стили

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Глобальные стили</title>
    <style type="text/css">
      h1 {
        font-size: 120%;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #333366;
      }
    </style>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

# Обзор клиентских технологий: CSS

## Внутренние стили

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Внутренние стили</title>
  </head>
  <body>
    <p style="font-size: 120%; font-family: monospace;
    color: #cd66cc">Пример текста</p>
  </body>
</html>
```

# Обзор клиентских технологий: JavaScript



- прототипно-ориентированный сценарный язык программирования
- скриптовый язык, предназначенный для создания интерактивных веб-страниц





# Обзор клиентских технологий: JavaScript

## Расположение JavaScript кода внутри страницы

```
<!DOCTYPE HTML>
<html>
<body>
  <p>Начало документа...</p>

  <script>
    alert('Привет, Мир!');
  </script>

  <p>...Конец документа</p>
</body>
</html>
```

Когда браузер читает HTML-страничку, и видит `<script>` - он первым делом читает и выполняет код, а только потом продолжает читать страницу дальше.

# Обзор клиентских технологий: JavaScript

## Расположение JavaScript кода в заголовке HEAD

```
<html>
  <head>
    <script type="text/javascript">
      function count_rabbits() {
        for(var i=1; i<=3; i++) {
          alert("Из шляпы достали "+i+" кролика!")
          // оператор + соединяет строки
        }
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="count_rabbits()"
      value="Считать кролей!"/>
  </body>
</html>
```

# Обзор клиентских технологий: JavaScript

## Расположение JavaScript кода во внешнем файле

`<script type="text/javascript" src="/Путь/к/script.js"></script>`

```
<html>
  <head>
    <script src="rabbits.js"></script>
  </head>
  <body>
    <script>
      count_rabbits();
    </script>
  </body>
</html>
```

### Содержимое файла rabbits.js:

```
function count_rabbits() {
  for(var i=1; i<=3; i++) {
    alert("Кролик номер "+i)
  }
}
```

# Обзор клиентских технологий: AJAX

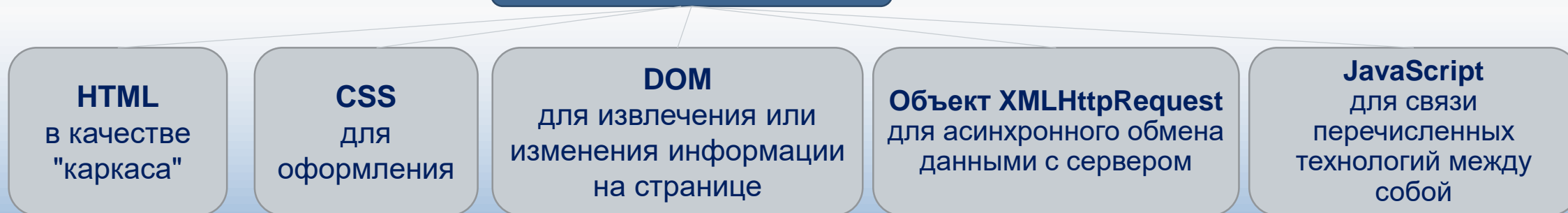
Подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.



Не новый язык программирования или разметки!

AJAX — не самостоятельная технология, а **концепция использования нескольких смежных технологий** - эффективный способ совместного использования **HTML, CSS, JavaScript и DOM**.

## AJAX использует



# Обзор серверных технологий: PHP



**PHP (Hypertext Preprocessor)** — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

Мощный инструмент для быстрого создания динамических и интерактивных веб-страниц.

Один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

Output Result:

My first PHP script!

# Обзор серверных технологий: PHP

## PHP файл:

PHP файл может **содержать** текст, HTML, CSS, JavaScript, и PHP код.

PHP код **исполняется на сервере**, а **результат** возвращается в **браузер** в виде обычного **HTML документа**.

Расширение файлов «**.php**»

PHP скрипт начинается с обозначения **<?php** и заканчивается **?>**:

```
<?php  
    // PHP code goes here  
?>
```

# Обзор серверных технологий: SQL



**SQL (Structured Query Language — структурированный язык запросов)**  
универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

- Информационно-логический язык.
- SQL нельзя назвать языком программирования.
- SQL остаётся единственным механизмом связи между прикладным программным обеспечением и базой данных. В то же время, современные СУБД, а, также, информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

## Преимущества

- Независимость от конкретной СУБД
- Наличие стандартов
- Декларативность

## Недостатки

- Несоответствие реляционной модели данных
- Сложность
- Отступления от стандартов
- Сложность работы с иерархическими структурами

# Обзор серверных технологий: ASP, ASP.NET



**ASP (Active Server Pages)** — активные серверные страницы



Первая технология компании Microsoft, позволяющая **динамически создавать веб-страницы на стороне сервера**.

Работает на операционных системах линейки **Windows NT** и на веб-сервере **Microsoft IIS**.

Не является языком программирования — это **технология предварительной обработки**, позволяющая **подключать программные модули во время процесса формирования веб-страницы**.

Относительная популярность ASP основана на **простоте используемых языков сценариев (VBScript или JScript) и возможности использования внешних COM-компонентов**.

Технология ASP получила своё развитие в виде **ASP.NET** — новой технологии создания веб-приложений, основанной на платформе **Microsoft .NET**.  
В начале 2011 технология **ASP** заменена **ASP.NET**.



# Обзор серверных технологий: Java



**Java (Джава) — объектно-ориентированный язык программирования.**

**Приложения Java** обычно компилируются в специальный байт-код, поэтому они **могут работать на любой виртуальной Java-машине (JVM)** вне зависимости от компьютерной архитектуры.

**Полная независимость байт-кода** от операционной системы и оборудования.

**Java** можно отнести к **клиентским технологиям**, так как Java-апплеты могут интерпретироваться на стороне клиента.

**Гибкая система безопасности** - исполнение программы полностью контролируется виртуальной машиной.

Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

# Рекомендованная литература:

- <http://www.google.com/>
- <http://htmlbook.ru/>
- <http://www.w3schools.com/>
- <https://www.w3.org/>