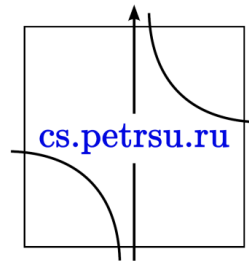


Введение

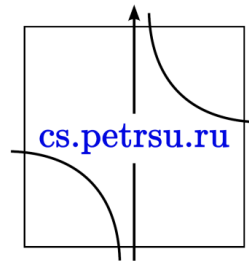
Глава №0

Технологии глобальной сети *World Wide Web*

- Структура и принципы Веб (базовые понятия, архитектура, стандарты и протоколы);
- Технологии сети Веб (языки разметки и программирования веб-страниц, инструменты разработки и управления веб-контента и приложений для Веб).



- *Сеть Веб* представляет собой глобальное информационное *пространство*, основанное на физической инфраструктуре Интернета и протоколе передачи данных *HTTP(s)*.
- *Сеть ARPANET*, министерства обороны США , 1969 г., скорость 50 Кбит/с. (Калифорнийский университет в Лос-Анджелесе, Стэнфордский исследовательский центр, Университет Юты и Университет штата Калифорния в Санта-Барбаре.)

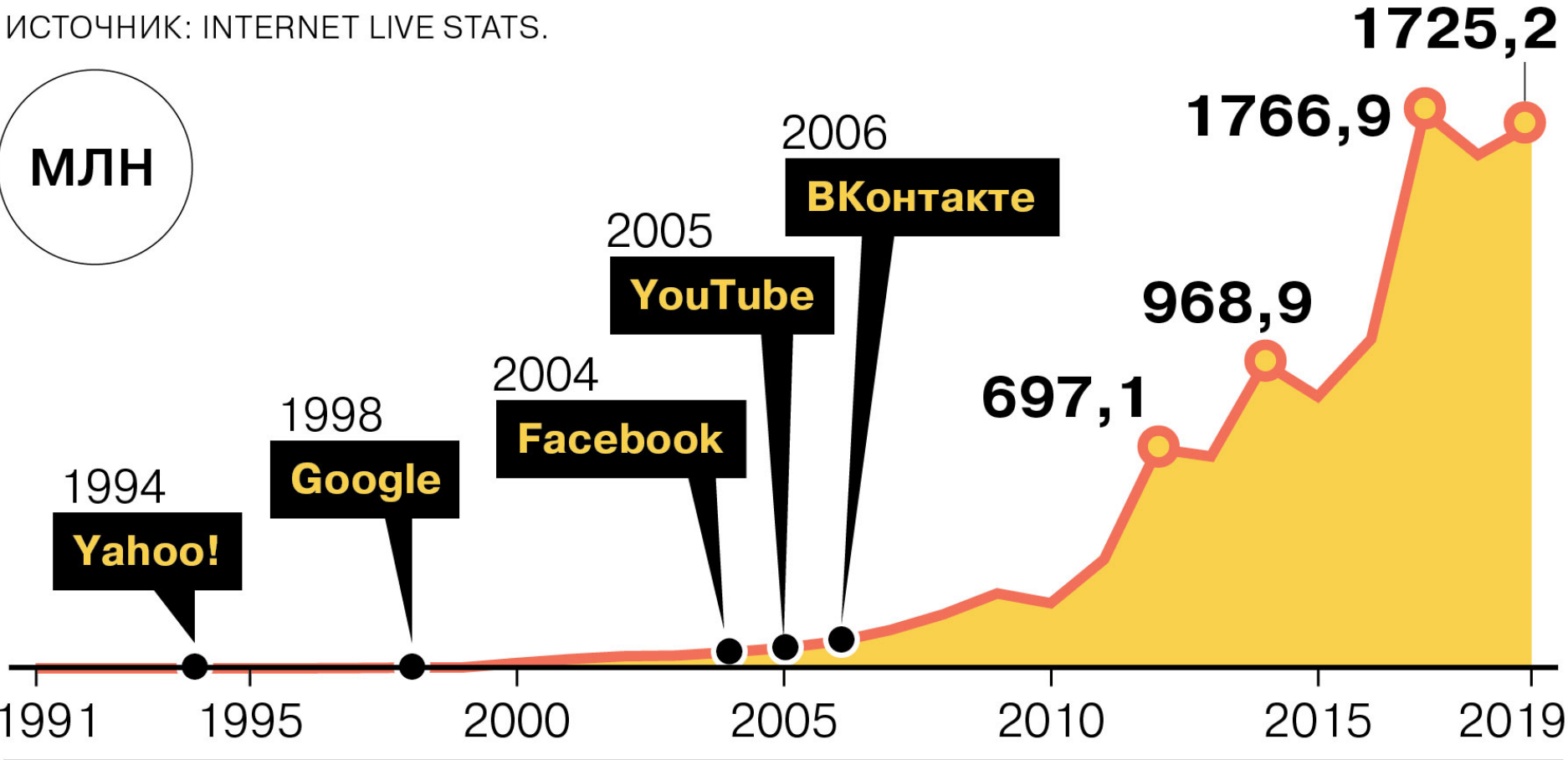


Год	Событие
1966	Эксперимент с коммутацией пакетов управления <i>ARPA</i>
1969	Первые работоспособные узлы сети <i>ARPANET</i>
1972	Изобретение распределенной электронной почты
1973	Первые компьютеры, подключенные к сети <i>ARPANET</i> за пределами США
1975	Сеть <i>ARPANET</i> передана в ведение управления связи мин. обороны США
1980	Начинаются эксперименты с TCP/IP
1981	Каждые 20 дней к сети добавляется новый хост
1983	Завершен переход на TCP/IP
1984	Запущена система DNS
1986	Создана <i>NSFnet</i> (компьютерная сеть Национального фонда науки США)
1989	Появление HTML, начало HTTP
1990	Сеть <i>ARPANET</i> прекратила существование
1991	Появление <i>Gopher</i> (протокол распределённого поиска)
1992	Публикация HTTP, появление Всемирной паутины.
1993	Первый графический браузер NCSA Mosaic
1995	Приватизация магистрали Интернета
1995	Yahoo, Hotmail, Amazon.com
1998	Число зарегистрированных доменных имен превысило 2 млн., Mail.ru
2000	Количество индексируемых веб-страниц превысило 1 млрд.

КАК РОСЛО КОЛИЧЕСТВО ВЕБ-САЙТОВ В МИРЕ

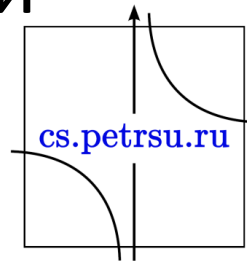
ИСТОЧНИК: INTERNET LIVE STATS.

МЛН



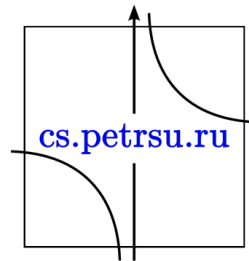
Консорциум W3C (World Wide Web Consortium)

- Организация, разрабатывающая и внедряющая технологические стандарты для Интернета и WWW.
- Основан Тимом Бернерсом-Ли в 1994 году (руководит по настоящее время).
- Единые принципы и стандарты, названы "*Рекомендациями*", которые затем внедряются разработчиками программ и оборудования.



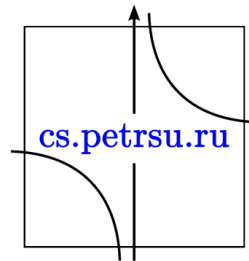
Эволюция сайтов

- Web 1.0 – Статичное содержание страниц, аскетичный дизайн, чаты, форумы, гостевые книги.
- Web 2.0 – новое поколение сайтов (после 2001) User-generated content. Разработка и использование API. RSS. Обновление страниц на лету (ajax).
- Web 3.0 – Community-generated content. Семантическая паутина. Уникальный идентификаторы и микроформаты. Аскетичный дизайн.



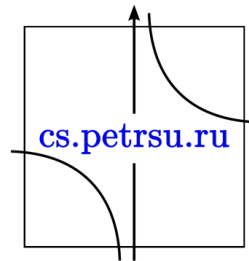
Тенденции развития ПО

- Клиент-серверная архитектура
- Глобальные приложения: поиск, почта – big data.
- Мобильные приложения
- ПО как сервис
- Облачные сервисы
- Тонкие клиенты
- Социальные сети



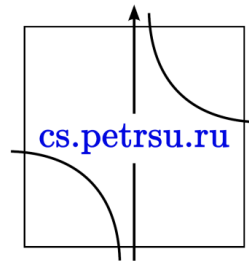
Web-приложения

- Традиционные сайты: новости, блоги, wiki, базы знаний, визитки.
- Глобальные приложения: почтовые сервисы, поиск, социальные сети.
- E-commerce: магазины, бронирование.
- Замена desktop приложениям: банк-клиенты, CRM, корпоративный софт.
- SAAS – различные сервисы и API



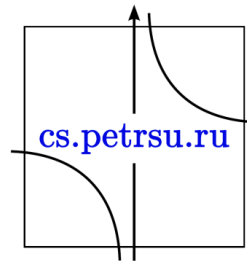
Направления деятельности

- Front-end разработка
- Back-end разработка
- Разработка под мобильные платформы
- Инфраструктура
- Архитектура
- Системное программирование
- Прочее (seo, security, management)



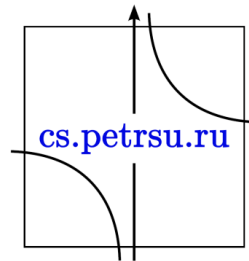
Протоколы TCP/IP

- Протокол *IP (Internet Protocol)* представляет собой протокол нижнего (сетевого) уровня и отвечает за передачу пакетов данных в сети (*датаграмм*, работает без подтверждений).
- *Протокол передачи данных TCP (Transmission Control Protocol)* (транспортного) уровня – отвечает за отправку и принятие пакетов через протокол *IP*, он гарантирует доставку всех переданных пакетов данных в правильной последовательности.



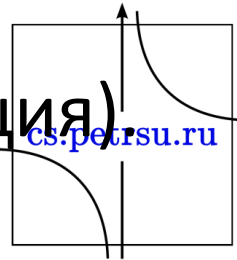
TCP/IP

- Для идентификации сетевых интерфейсов используются 3 типа адресов:
 - аппаратные адреса (или MAC-адреса);
 - сетевые адреса (IP-адреса);
 - символьные (доменные) имена.

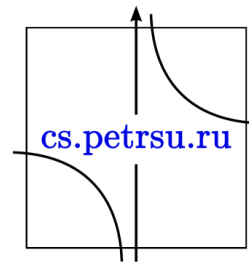


Система доменных имен DNS

- Пользователи предпочитают *символьные имена* (*доменные имена*).
- В сетях *TCP/IP* используется *доменная система имен*, имеющая иерархическую (в виде дерева) структуру (международный стандарт *ISO 3166*).
- Для обозначения стран используются двухбуквенные аббревиатуры, например *ru* (Российская Федерация), *us* (США), *it* (Италия), *fr* (Франция).



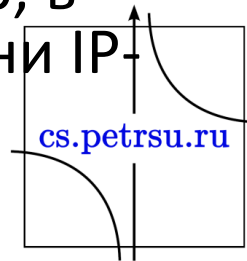
- Для различных типов организаций используются трехбуквенные аббревиатуры:
 - *net* – сетевые организации;
 - *org* – некоммерческие организации;
 - *com* – коммерческие организации;
 - *edu* – образовательные организации;
 - *gov* – правительственные организации.



Существуют две схемы разрешения *DNS*-имен

Нерекурсивная процедура:

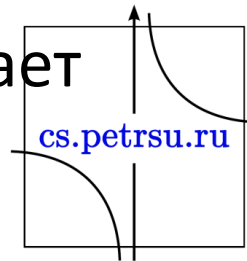
- DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени;
- DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части имени;
- DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т.д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Сервер дает окончательный ответ клиенту.



Существуют две схемы разрешения *DNS*-имен

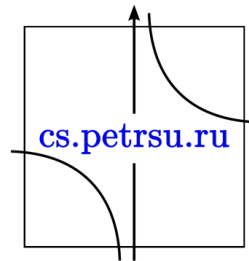
Рекурсивная процедура:

- DNS-клиент запрашивает локальный DNS-сервер, обслуживающий поддомен, которому принадлежит клиент;
- Далее
- Если локальный DNS-сервер знает ответ, он возвращает его клиенту
- Если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу. После получения ответа сервер передает его клиенту.

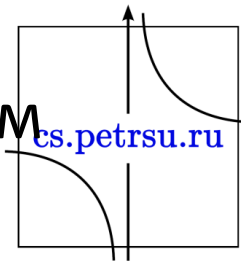


Структура и принципы WWW

- **Веб-сервер** является программой, запускаемой на подключенном к сети компьютере и передающей данные по протоколу *HTTP*.
- Для идентификации ресурсов (зачастую файлов или их частей) в *WWW* используются идентификаторы ресурсов **URI** (*Uniform Resource Identifier*).
- Для определения местонахождения ресурсов в этой сети используются локаторы ресурсов **URL** (*Uniform Resource Locator*). - Комбинация *URI* и системы *DNS*.



- **Веб-браузер** - специальная программа на клиентском компьютере для просмотра информации, полученной от веб-сервера.
- Основная функция веб-браузера - *отображение* гипертекстовых страниц (веб-страниц).
- Для создания гипертекстовых страниц в WWW используется язык *HTML*.
- Множество веб-страниц образуют **веб-сайт**.
- **Прокси-сервер** (proxy-server) — служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам.

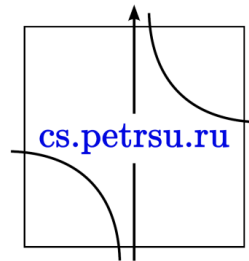


"Классическая" схема *HTTP*-сеанса:

- Установление *TCP*-соединения.
- Запрос клиента.
- Ответ сервера.
- Разрыв *TCP*-соединения.

В состав *HTTP*-запроса, передаваемого клиентом серверу, входят следующие компоненты.

- Строка состояния.
- Поля заголовка.
- Пустая строка.
- Тело запроса.



HTTP

- Запрос:

GET /wiki/страница HTTP/1.1

Host: ru.Wikipedia.org

Accept: text/html

Connection: close

(пустая строка)

- Ответ:

HTTP/1.1 200 OK

Server: Apach

Content-Language: ru

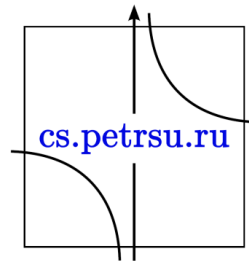
Content-Type: text/html; charset=utf-8

Content-Length: 1234

Connection: close

(HTML)

TelNet

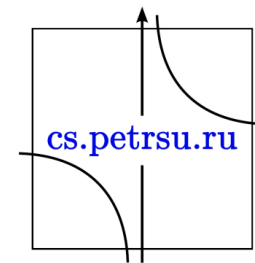


Поля заголовка	Значение
Host	Доменное имя или IP-адрес узла, к которому обращается клиент
Referer	URL документа, который ссылается на ресурс, указанный в строке состояния
From	Адрес электронной почты пользователя, работающего с клиентом
Accept	MIME-типы данных, обрабатываемых клиентом. Это поле может иметь несколько значений, отделяемых одно от другого запятыми. Часто поле заголовка Ассепт используется для того, чтобы сообщить серверу о том, какие типы графических файлов поддерживает клиент
Accept-Language	Набор двухсимвольных идентификаторов, разделенных запятыми, которые обозначают языки, поддерживаемые клиентом
Accept-Charset	Перечень поддерживаемых наборов символов
Content-Type	MIME-тип данных, содержащихся в теле запроса (если запрос не состоит из одного заголовка)
Content-Length	Число символов, содержащихся в теле запроса (если запрос не состоит из одного заголовка)
Range	Присутствует в том случае, если клиент запрашивает не весь документ, а лишь его часть
Connection	Используется для управления TCP-соединением. Если в поле содержится Close, это означает, что после обработки запроса сервер должен закрыть соединение. Значение Keep-Alive предлагает не закрывать TCP-соединение, чтобы оно могло быть использовано для последующих запросов
User-Agent	Информация о клиенте

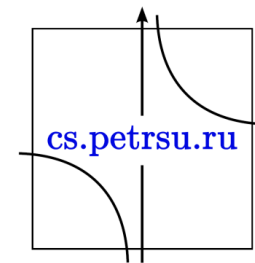


HTML

- HTML (HyperText Markup Language — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине.
 - HTML 0.9
 - HTML 2.0, одобренный как стандарт 22 сентября 1995 года;
 - HTML 3.2— 14 января 1997 года;
 - HTML 4.0 — 18 декабря 1997 года;
 - HTML 4.01 — 24 декабря 1999 года;
 - HTML 5 — 28 октября 2014 года
 - HTML 5.1 – 1 ноября 2016 года.
 - HTML 5.2 – 14 декабря 2017 года.
 - HTML 5.3 – 24 декабря 2018 года.



- HTML — теговый язык разметки документов.
- Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — **тегами**.
- Элементы могут быть пустыми (не содержат никакого текста и данных). В этом случае обычно не указывается закрывающий тег (напр., тег переноса строки `
` — одиночный, закрывать не нужно).
- Элементы могут иметь **атрибуты**, определяющие какие-либо их свойства (напр., атрибут `href=""` у ссылки). Атрибуты указываются в открывающем теге.
- Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения **не имеет**.
- Элементы могут быть вложенными.



Структура html-документа

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>HTML Document</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>
```

```
      <b>
```

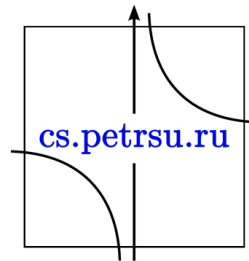
Этот текст будет полужирным, *<i>а этот — ещё и курсивным</i>*.

```
      </b>
```

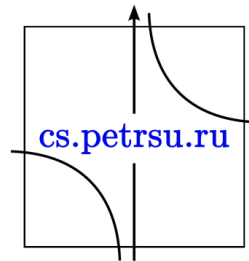
```
    </p>
```

```
  </body>
```

```
</html>
```



- **HTML** — элемент верхнего уровня — ограничивает начало и конец документа. Элемент HTML является элементом верхнего уровня в языке HTML. Использование элемента HTML не сказывается на выводе документа. Если элемент HTML используется, начальный и конечный теги HTML должны идти вокруг всего документа, сразу после объявления DOCTYPE.
- Основная структура HTML-документа должна состоять из заголовка и тела.
- **HEAD** — заголовок документа. В документе должен присутствовать только один элемент HEAD, и он должен появиться до элемента BODY .
 - ISINDEX; META; LINK; STYLE; SCRIPT.

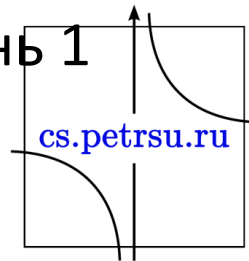


- **BODY** — тело документа. Если элемент BODY не содержит атрибуты, использование его не дает явного эффекта в непосредственном отображении документа. Только один элемент BODY разрешен в документе, и он должен находиться после элемента HEAD.

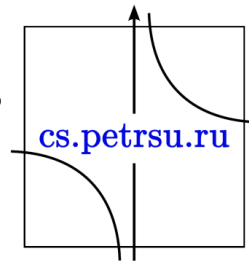
Имя атрибута	Значения	Смысл
BGCOLOR	Цвет	Фоновый цвет документа
TEXT	Цвет	Цвет для текста документа
LINK	Цвет	Цвет для непосещенной гипертекстовой связи
VLINK	Цвет	Цвет для посещенной гипертекстовой связи
ALINK	Цвет	Цвет для активной гипертекстовой связи; используется для выделения текста , когда пользователь нажал на связь
BACKGROUND	URL	URL фона

Теги управления форматированием

- Абзац формируется при помощи тега `<p>...</p>`.
 - Изначально предполагалось использовать вместе с тегами параметры. Параметры тега – способы выравнивания (`left`, `right`, `center`, `justify`). Например, если необходимо сделать абзац с выравниванием по ширине, то надо написать:
- Заголовки предназначены показать важность раздела, к которому относятся, с помощью различных заголовков легко регулировать размер текста.
 - Чем выше уровень заголовка, тем больше размер шрифта. Самым верхним уровнем является уровень 1 (`<h1>`), а самым нижним — уровень 6 (`<h6>`).

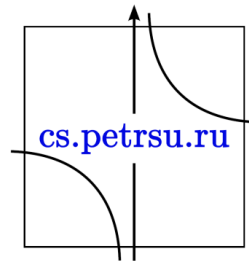


- Полужирное начертание: `` и ``
- Курсивное начертание: `<i>` и ``
- Маркированный список: формируется с помощью контейнера ``, а каждый пункт списка начинается с тега ``.
 - В списке непременно должен присутствовать закрывающий тег ``, иначе возникнет ошибка. Закрывающий тег `` хотя и не обязателен, но советуем всегда его добавлять, чтобы четко разделять элементы списка.
- Нумерованные списки: `` и элемент ``.



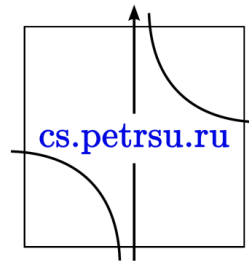
Таблицы

- Таблицы могут использоваться для наглядного предоставления информации и для верстки веб-страниц.
- Таблица с невидимой границей представляет собой словно модульную сетку, в блоках которой удобно размещать элементы веб-страницы.
- В настоящее время верстка страниц в осуществляется с помощью слоев.



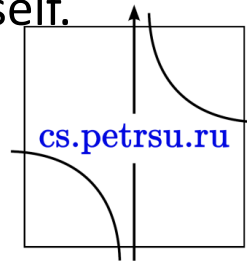
Пример таблицы

```
<table border="1">  
  <tr>  
    <td colspan="2">1</td>  
  </tr>  
  <tr>  
    <td>3</td>  
    <td>4</td>  
  </tr>  
</table>
```

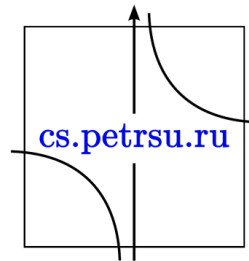


Ссылки

- Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую.
- Особенность их состоит в том, что сама ссылка может вести не только на HTML-страницы, но и на файл любого типа, причем этот файл может размещаться совсем на другом ресурсе.
- `ключевая фраза`
 - `_blank` — загружает страницу в новое окно браузера.
 - `_self` — загружает страницу в текущее.
 - `_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`.
 - `_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`.

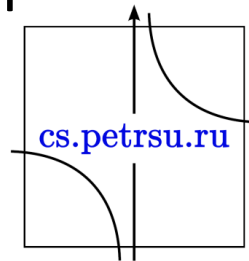


- Ссылки на адрес электронной почты:
`Написать мне`
- Кроме внешних ссылок можно использовать внутренние ссылки (якорь):
 - `` - установка якоря.
 - `В начало` - ссылка на якорь.



Рисунки

- Для вставки изображения применяется тег ``, атрибут `src` определяет адрес графического файла.
- Атрибуты: `alt`, `width`, `height`.
 - `alt` предназначен для создания альтернативного текста (будет отображаться если изображение не загрузилось).
 - `width` и `height` задается размер. Заданный размер не влияет на скорость загрузки, его определяет реальный размер изображения.



Карты изображения

- Карты изображения позволяют привязывать ссылки к разным областям одного изображения. По сути это графические изображения с размеченными областями.

```

```

- Далее необходимо определить области на карте следующим образом:

```
<map name="map1">
```

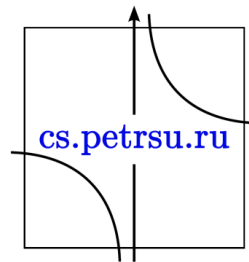
```
  <area shape="форма" coords="список координат" href="ссылка">
```

```
  <area shape="poly" coords="282,2, 222,1, 203,6, 210,27, 212,33, 282,31, 301,33, 301,28" href="1.html">
```

```
  ...
```

```
</map>
```

- rect - прямоугольник (координаты двух вершин)
- poly - многоугольник (координаты всех вершин)
- circle - окружность (координаты центра и радиус)



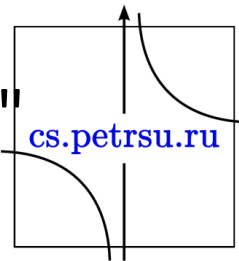
Фреймы

```
<FRAMESET ROWS="45%,45%,*">  
  <FRAME SRC="B611.HTM">  
  <FRAME SRC="B612.HTM">  
  <FRAME SRC="B613.HTM">  
</FRAMESET>
```

В настоящее время фреймы практически не используются.

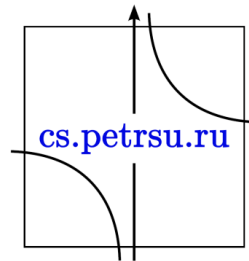
Активно используются плавающие фреймы используя тэг <IFRAME>

```
<iframe src="111.html" name="iframe1" width="300"  
height="300" align="left"></iframe>
```



Другие часто используемые теги:

- `<div>` - слой, популярный блочный элемент.
- `` - предназначен для определения строчных элементов документа.
- `` - контейнер для изменения характеристик шрифт, считается устаревшим.
- `<tt>` отображает текст моноширинным текстом
- `<strike>`, `<sub>`, `<sup>`
- `<form>`
- `<input>`
- `<textarea>`
- `<select>`



Разработка и отладка

The screenshot shows a web browser window with the URL `http://htmlbook.ru`. The page content is titled "Шпаргалки про телеграм-бота" (Telegram bot cheat sheet). The main text describes bot development, mentioning JavaScript, Node.js, and Telegram. It includes a list of links for further reading: "Как сделать бота на JavaScript", "Как загрузить его на сервер", and "Как сделать, чтобы бот работал с Гугл Таблицами".

On the left, there is a sidebar with a "Справочник CSS" (CSS Reference) section, listing various CSS properties like `!important`, `-moz-border-bottom-colors`, etc.

At the bottom, the browser's developer tools are open. The "Elements" panel shows the DOM tree with the following structure:

```
<script type="text/javascript" src="//an.yandex.ru/system/context.js" async></script>
<script type="text/javascript" src="//an.yandex.ru/system/context.js" async></script>
<script type="text/javascript" src="https://htmlbook.ru/.well-known/ddos-guard/check"
  async defer></script>
<header>...</header>
<form action="/example/" method="post" id="codeform">...</form>
<div class="dialog-off-canvas-main-canvas" data-off-canvas-main-canvas>
  <div class="layout"> == $0
    <aside>...</aside>
    <div class="block" id="block-htmlbook-block-16">...</div>
    <div id="sidebar">
      <div class="block" id="block-htmlbook-block-25">...</div>
      <div class="block" id="block-views-block-popular-faq">...</div>
```

The "Styles" panel shows the default user agent styles for a `div` element:

```
element.style {
}
.layout {
  overflow: hidden;
  padding-bottom: 30px;
  background: url(/themes/hb/img/bg.png) repeat-y;
  background: linear-gradient(to right, #4b4a45, #4b4a45 210px, transparent 211px, transparent);
}
div {
  user agent stylesheet
```

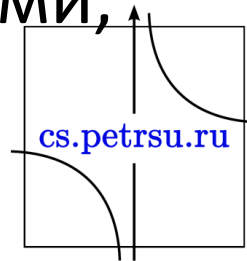
Верстка, типы макетов

- Фиксированные – контент располагается по центру окна браузера, его ширина ограничивается размерами в пикселах;
- Резиновые - задается в процентах ширина колонок так, что макет занимает всю свободную ширину окна браузера;
- Эластичные - размер элементов задается в em, привязанных к размеру шрифта;
- Адаптивные - подстраивается под разрешение монитора и окна браузера, меняя ширину макета, число колонок, размеры изображений и текста;
- Комбинированные - предполагает использование разной ширины для отдельных частей страницы (шапку и подвал делают резиновыми, контент фиксированным).



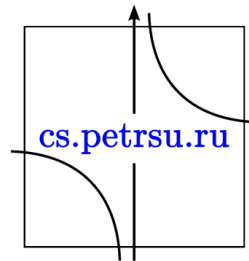
Каскадные таблицы стилей

- Таблицы стилей (или каскадные таблицы стилей, CSS) – это описание правил, задающих параметры представления отдельных элементов на языке HTML.
- CSS появились одновременно с HTML 4.0 (Dynamic HTML). Сам термин «каскадные таблицы стилей» был предложен в 1994 году. Все объявления CSS называются селекторами, записываются в фигурных скобках.



Размещение каскадных таблиц

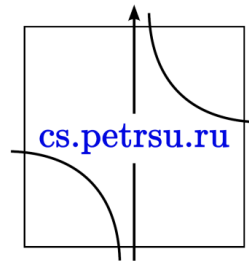
- Свойства можно определять непосредственно с тегом.
 - Например, вставка следующего тега сделает текст красным на синем фоне `красный на синем`
- Таблицу можно разместить между тегами `<style>` и `</style>`
- Таблицу стилей можно разместить в отдельном файле.
 - Подключение затем выполняется так: `<link rel="stylesheet" type="text/css" href="style.css">`



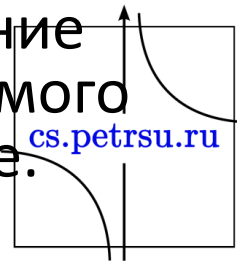
Параметры CSS, управляющие положением на странице

position: absolute | fixed | relative | static | inherit

- Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице. **absolute** указывает, что используются абсолютные координаты. **relative** указывает, что используются относительные координаты. **inherit** наследует значение родителя.

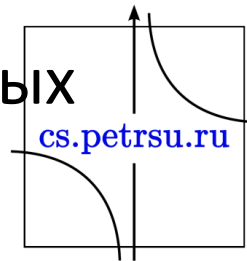


- **left** задает положение относительно левого края контейнера.
- **top** задает положение относительно верхнего края контейнера. Задаются в процентах или пикселях.
- **z-index** указывает на то, какой элемент должен располагаться выше при перекрытии. Измеряется в единицах.
- **visibility** определяет видимость элемента. Принимает значения `visible` (видимый), `hidden` (скрытый), `inherit` (наследуется от родительского элемента).
- **margin** устанавливает величину отступа от каждого края элемента. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.
- **padding** устанавливает значение полей вокруг содержимого элемента. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое.

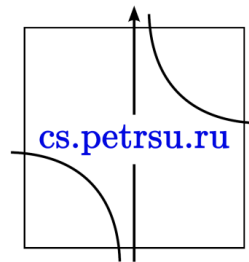


Правила построения CSS

- Каждое правило CSS из файла имеет две основные части — *селектор* и *блок объявлений*.
- **Селектор**, расположенный в левой части правила до знака «{», определяет, на какие части документа (возможно, специально обозначенные) распространяется правило.
- **Блок объявлений** располагается в правой части правила. Он помещается в фигурные скобки, и состоит из одного или более *объявлений*, разделённых знаком «;». Каждое *объявление* представляет собой сочетание *свойства CSS* и *значения*, разделённых знаком «:».

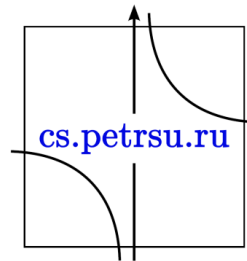


```
селектор, селектор {  
    свойство: значение;  
    свойство: значение;  
    свойство: значение;  
}
```



Виды селекторов

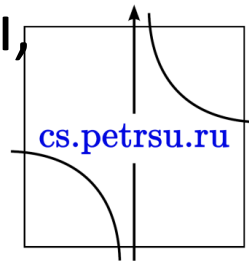
- Универсальный селектор
- Селектор тегов
- Селектор классов
- Селектор идентификаторов
- Селектор атрибутов
- Селектор потомков
- Селектор дочерних элементов
- Селектор элементов одного уровня (смежные и соседние)
- Селектор псевдоклассов
- Селектор псевдоэлементов



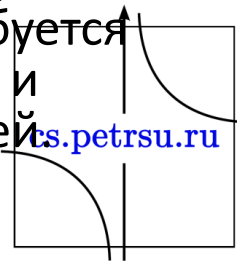
Наследование. Каскадирование. Приоритеты стилей CSS.

Применение CSS к документам HTML основано на принципах *наследования* и *каскадирования*.

- Принцип **наследования** заключается в том, что свойства CSS, объявленные для элементов-предков, почти всегда наследуются элементами-потомками.
- Принцип **каскадирования** применяется в случае, когда какому-то элементу HTML одновременно поставлено в соответствие более одного правила CSS, то есть, когда происходит конфликт значений этих правил. Чтобы разрешить такие конфликты, вводятся правила приоритета.



- Наиболее низким приоритетом обладает стиль браузера;
- Следующим по значимости является стиль, заданный пользователем браузера в его настройках;
- И наиболее высоким приоритетом обладает стиль, заданный непосредственно автором страницы. Далее приоритеты расставляются следующим образом:
 - Самым низким приоритетом обладают стили, наследуемые в документе элементом от своих предков;
 - Более высоким приоритетом обладают стили, заданные во внешних таблицах стилей, подключённых к документу;
 - Ещё более высоким приоритетом обладают стили, заданные непосредственно селекторами всех десяти видов, содержащимися в контейнерах `style` данного документа. Когда к элементу имеют отношение несколько таких селекторов - конфликты разрешаются с помощью расчёта **специфичности** каждого такого селектора и применения этих селекторов к данному элементу в порядке убывания их специфичностей.
- (!) Самым высоким приоритетом обладают стили, объявленные с помощью сопроводительного правила `!important`. Если таких свойств несколько, то предпочтение отдаётся в первую очередь стилям, заданным автором или пользователем, а для остальных свойств, которые будут являться задаваемыми автором страницы, потребуется определить их специфичности по принципам, описанным выше, и применять эти свойства в порядке убывания этих специфичностей.



Специфичность селекторов

Делится на 4 группы — a, b, c, d:

- если стиль встроенный (определён как `style="..."`, то $a=1$, иначе $a=0$);
- значение b равно количеству идентификаторов (иначе — `id=" "`, они начинаются с `#`) в селекторе;
- значение c равно количеству классов (`class=" "`, они начинаются с `.`), псевдоклассов (они начинаются с `:`, например `a:hover`) и селекторов атрибутов (`input[type="text"]`);
- значение d равно количеству селекторов типов элементов. После этого полученное значение приводится к числу (обычно в десятичной системе счисления).

Селектор, обладающий большим значением специфичности, обладает и большим приоритетом.

Таблица расчёта специфичности:

Селектор	a, b, c, d	Число
<code>span</code>	0, 0, 0, 1	1
<code>div .class</code>	0, 0, 1, 1	11
<code>#id .class</code>	0, 1, 1, 0	110
<code>div span</code>	0, 0, 0, 2	2
<code>.class</code>	0, 0, 1, 0	10
<code>#id span</code>	0, 1, 0, 1	101