

Оснoвы WEB

Интернет и Всемирная паутина

Интернет — всемирная система объединённых компьютерных сетей для хранения и передачи информации.

На основе интернета работает “Всемирная паутина” и множество других систем передачи данных.

Всемирная паутина (World Wide Web, WWW) — распределённая система, предоставляющая доступ к связанным (по гиперссылкам) между собой документам, расположенным на различных компьютерах, подключённых к Интернету.

Первый “запуск” сети интернет произошел в США (Калифорнийский университет в Лос-Анджелесе), в 1969 году. Изначально он объединял научные учреждения.



Интернет в 2016 году



Сетевой протокол

— набор правил и действий (очерёдности действий), позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами.

Интернет сеть построена на базе стека протоколов TCP/IP.

TCP/IP

4

Прикладной

Множество протоколов прикладного уровня, выполняющих конкретные прикладные задачи: **HTTP, FTP, SMTP, SSH, DNS** и др.

3

Транспортный

Протокол TCP используется для передачи данных.
Контролирует длину сообщения, скорость обмена, трафик, осуществляет повторный запрос данных в случае потери и устраняет дублирование.

2

Сетевой

Протокол IP описывает структуру сети и доставку **пакетов**.
Но не гарантирует надёжной доставки пакета до адресата.

1

Канальный

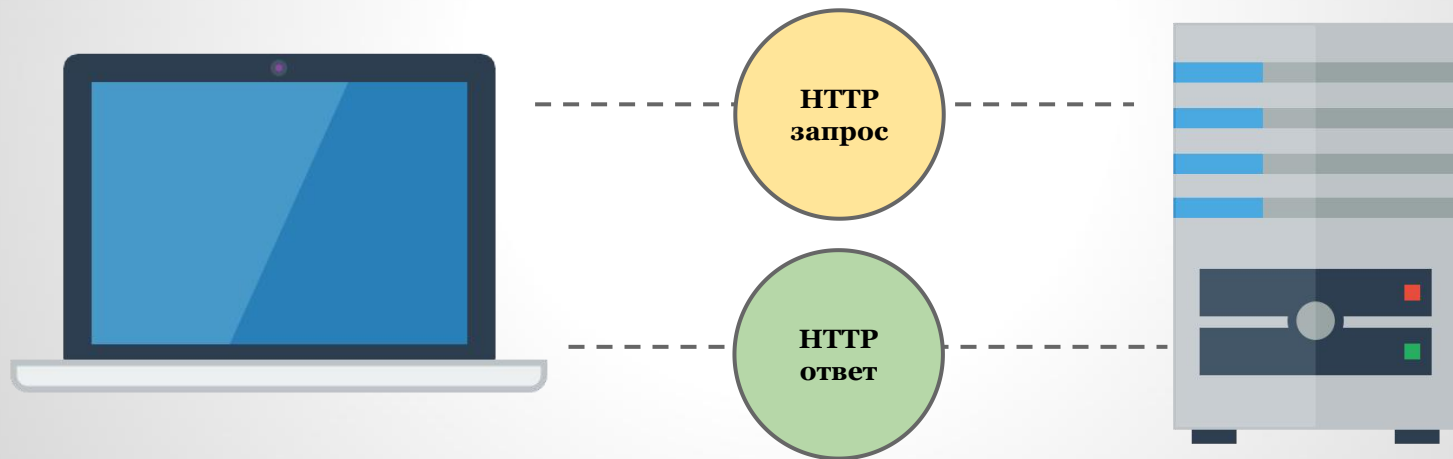
Физическая среда и принципы кодирования информации. *Например, интерфейс Ethernet, описывающий передачу данных по коаксиальному кабелю или витой паре.*

TCP/IP vs OSI

TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTP, Telnet, NTP, DHCP, PING	Application
		Presentation
		Session
Transport	TCP, UDP	Transport
Network	IP, ARP, ICMP, IGMP	Network
Network Interface	Ethernet	Data Link
		Physical

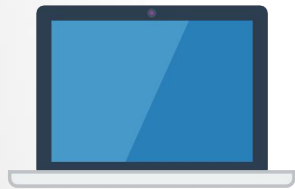
HTTP: HyperText Transfer Protocol

Протокол прикладного уровня передачи данных. Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Universal Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы.



Клиент-сервер

— это сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми **серверами**, и заказчиками услуг, называемыми **клиентами**.



Клиент



Сервер

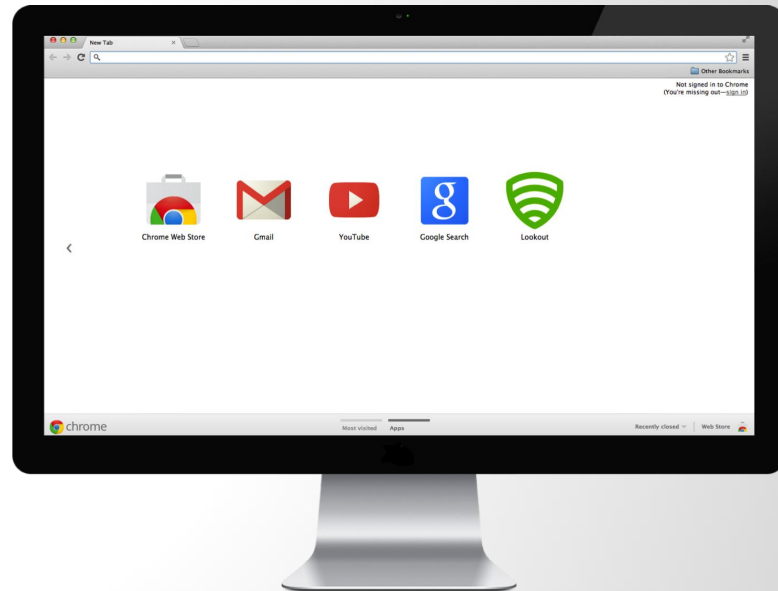
Клиент, веб-браузер и frontend

Для просмотра информации, полученной от веб-сервера используется **браузер**.

Основные функции

- установка связи с веб-сервером, на котором хранится документ, и загрузка всех его компонентов;
- форматирование и отображение веб-страниц;
- поиск веб-страниц и упрощение доступа к страницам, посещавшимся раньше;

Frontend — интерфейс взаимодействия между пользователем и **backend**.



Сервер, веб-сервер и backend

Сервер - это компьютер, с установленными на нём программами, выполняющими различные (серверные) задачи. Одна из таких программ, называется “веб-сервер”.

Веб-сервер — это компьютерная программа, запускаемая на подключённом к сети компьютере и использующая протокол HTTP для передачи данных.

Часто, в целях упрощения, термины “**веб-сервер**” и “**сервер**” одновременно подразумевают и программу и сам компьютер, на котором она установлена.

backend - это серверная сторона сайта, написанная на серверных языках программирования: **C#, java, node.js, python**, которые, в том числе, обеспечивают взаимодействие с базой данных.



Сетевой порт

В TCP пакетах указываются порт источника и порт назначения.

Сетевой порт — условное число от 1 до 65535, указывающее, какому приложению предназначается пакет.

В каждом пакете присутствуют IP адрес узла-источника и IP адрес узла-назначения.

По умолчанию в протоколе HTTP используется **порт 80**, а в протоколе HTTPS — **порт 443**

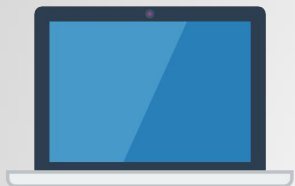
*IP адрес - почтовый адрес дома,
а порт - номер квартиры
конкретного жильца.*

DNS

— компьютерная **распределённая система** для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (доменному имени).

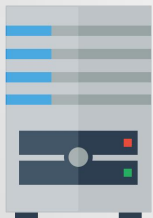


Клиент



1

https://goit.ua/



DNS

185.68.16.121



2

HTTP запрос: 185.68.16.121



3

HTTP ответ: HTML, CSS, JavaScript,
медиа-поток



Сервер



Программы на сервере

Веб-сервер (443 порт)
MySQL
PHP / Java
nginx
Предоставляется хостингом



ОСНОВНЫЕ ЧАСТИ

URI / URL

Адрес который мы набираем в браузере `https://goit.ua/`

HTTP METHOD

Последовательность из любых символов, указывающая на основную операцию над ресурсом `req.method`
`GET / POST / PUT / DELETE`

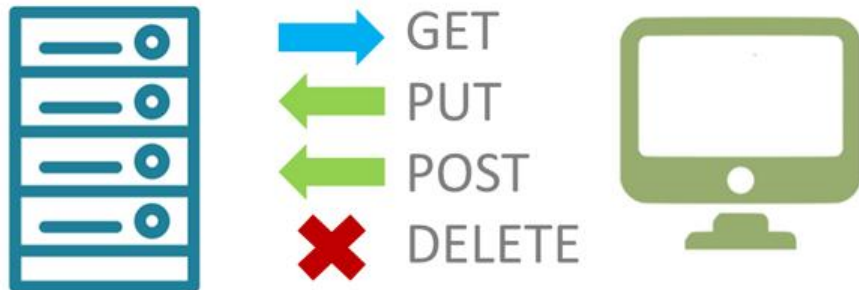
HTTP STATUS CODE

Код ответа сервера на запрошенную операцию
`200 OK, 404 Not Found`

HTTP-МЕТОДЫ

GET/POST/PUT/DELETE

Тип HTTP-запроса



- › **GET** запрашивает представление ресурса. Запросы с использованием этого метода могут только извлекать данные.
- › **POST** используется для отправки сущностей к определённому ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты на сервере.
- › **PUT** заменяет все текущие представления ресурса данными запроса.
- › **DELETE** удаляет указанный ресурс.
- › **PATCH** используется для частичного изменения ресурса.

ВЕБ-СТЕК

- это набор программного обеспечения, который обрабатывает HTTP-запросы и веб-страницы и возвращает их браузеру.

- **LAMP** - Linux, Apache, MySQL, (PHP , Python)
- **WAMP** - Windows, Apache, MySQL, (PHP , Python)
- **WISA** - Windows, IIS, MS SQL, .NET
- **MERN** - MongoDB, Express, React, Node.js

Хостинги

Облачный хостинг



Google Cloud



Виртуальный выделенный сервер (VPS/VDS)

- + гарантированная часть ресурсов сервера;
- + возможность устанавливать ПО.
- невысокая мощность из-за ограничения выделяемых ресурсов;
- забота о настройках и работе ПО переходит на сторону клиента;
- платная техническая поддержка;
- дороже виртуального хостинга.

**VPS и VDS - это разные названия одного и того же*

Домен

Как получить домен:

1. Получить в подарок при покупке хостинга
2. Купить у той же компании, которая предоставляет хостинг
3. Купить в другой компании

Кто продаёт домены (где брал)

1. <https://nic.ua/>
2. <https://www.hostinger.com.ua/>



HTML

HTML - декларативный язык программирования. Используемый для разметки документов в сети интернет. HTML “рендерится” (отображается, исполняется) в браузерах. Таким образом, может отображаться на всех устройствах (мобильные, планшеты, стационарные компьютеры и т.п) где есть установленный браузер.

СТРУКТУРА

```
<!doctype html> <!-- доктайп (указание спецификации) -->
<html lang="ru"> <!-- открывающий тег HTML -->
  <head>
    ... <!-- настройки документа, подключение стилей отображения -->
  </head>
  <body>
    ... <!-- тело документа -->
  </body>
</html>
```

Tag

ТЕГ - элемент разметки, может применяться с настройками (атрибутами) или без них. Текст или другой контент “внутри” тега отображается в зависимости от типа, и названия тега а также в зависят от того какие атрибуты “стоят” у тега.

```
<tagname attrname="attrprop"> tag content </tagname>
```

```
<tagname attrname="attrprop" />
```

```
<!--/----- параграф ----->
```

html

```
<p>
```

Я просто параграф

текста

```
</p>
```

```
<!--/----- список ----->
```

```
<ul>
```

```
<li>Пункт 1</li>
```

```
<li>Пункт 2</li>
```

```
<li>Пункт 3</li>
```

```
</ul>
```

браузер

Я просто параграф текста

- Пункт 1
- Пункт 2
- Пункт 3

<doctype> – указание спецификации HTML

<doctype> - создан для того что бы браузер понимал согласно какому стандарту и версии отображать веб-страницу.

пример: (<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" <http://www.w3.org/TR/html4/strict.dtd>>) - html 4.01

примечание: **W3C (World Wide Web Consortium)** — организация, разрабатывающая и внедряющая технологические стандарты интернета.

```
<!doctype html> <!-- доктайп (указание спецификации) -->
```

<html>

<html> - тег-контейнер для “головы” и “тела” документа. Больше никаких элементов внутри него лежать не должно.

примечание: без тега **<html>** документ всё равно отобразится, но написание его принято мировым сообществом, поэтому его применение **обязательно!**

<head>

<head> - тег, визуально не отображаемый (не рендерится) браузером. В этот тег вносятся настройки документа, метатеги, теги для распознавания seo, заголовки страницы, подключение стилей и т.п.

примечание: **search engine optimization, SEO** — комплекс мер для поднятия позиций сайта в [результатах выдачи поисковых систем](#) по определенным запросам пользователей с целью [продвижения сайта](#).

```
<head>  
... <!-- настройки документа, подключение стилей отображения -->  
</head>
```

<title>

<title> - тег, указывающий браузеру на название страницы. Которое отобразится в заголовке вкладки или строке состояния (у очень старых браузеров).

```
<head>  
  <title>Главная страница</title>  
</head>
```

html

кодировка документа

кодировка текст - это способ хранения символьной информации. Может выставляться как на сервере, так и прямо в документе. Текущий общепринятый стандарт - **UTF-8**

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

html

seo данные

мета-теги SEO - нужны для поисковых роботов. По ним робот считывает информацию о странице.

```
<head>  
  <meta name="description" content="Описание для сайта"/>  
  <meta name="keywords" content="сайт, приложение, html, css"/>  
  <meta name="author" content="Имя Фамилия"/>  
</head>
```

html

<body>

<body> - тег определяющий “тело” документа. Именно в этот тег помещаются вся остальная разметка и подключения скриптов JavaScript.

```
<body>  
    ... <!-- тело документа -->  
</body>
```

"комментарии"

- В разных языках есть разный синтаксис для написания комментариев.
- В HTML - это конструкция вида:

<!-- текст комментария или кусок кода -->

- Все что внутри конструкции никак не влияет на код и не отображается браузером.

```
<!-- я просто комментарий кода и в браузере я не отображаюсь -->
```

Для того что бы переходить от одного документа к другому существует тег ссылки - **<a> (Anchor Tag)**

```
<a href="http://google.com"> ссылка на гугл </a>
```

html

[ссылка на гугл](http://google.com)

браузер

Атрибуты

href - адрес куда ведет ссылка. Может быть сторонним ресурсом или локальным файлом (так же может быть указан идентификатор блока на странице до которого стоит прокрутить страницу).

title - добавляет всплывающую подсказку для ссылки.

download - указывает что файл по ссылке нужно не открыть, а скачать.

target - указывает в каком окне открыть ссылку (в текущем или в новом).

```
<a href="http://google.com" target="_blank"> ссылка на  
гугл </a>
```

html

Пути бывают **абсолютными** и **относительными**.

Абсолютный путь - ищет файл от “корневой папки” в зависимости от конфигурации сервера (начинается со слеша).

```
<a href="/pages/contacts.html">контакты</a>
```

html

Относительный путь - ищет файл от текущего файла где прописан путь (начинается с точек).

```
<a href="../pages/contacts.html">контакты</a>
```

html

Ссылка может “проскроллить” до нужного блока на странице. Это называется **переход по якорю**.

```
<a href="#block">Прокрутить к блоку</a>
```

html

```
<div id="block">Блок который нужно показать</div>
```

html

CSS

CSS (каскадные таблицы стилей) - этим языком описано то как будет представлен HTML. Как и html это декларативный язык, т.е в нем нет алгоритмической составляющей, а только описание вида или поведения указанных элементов.



Правила CSS всегда передаются в виде пары - название свойства - его значение, разделенной двоеточием. Каждое правило **обязательно** должно заканчиваться точкой с запятой.

```
key : value;
```

CSS

CSS селектор - указание к какому именно элементу или набору элементов будет применены стили.

```
a {  
  font-size: 16px;  
}
```

CSS

Существует 2 вида комментариев (блочные и строчные). В html и css они реализованы одной конструкцией.

CSS

```
/*  
    блочный комментарий  
    в CSS  
*/
```

Выборка элементов

Выборка по тегу

под этот селектор попадают все теги указанные в селекторе

html

```
<h1>
```

будет отстилизован

```
</h1>
```

```
<h1>
```

И этот тоже попадет

```
</h1>
```

```
<h3>
```

а этот останется не тронутым

```
</h3>
```

CSS

```
h1 {
```

```
/* стили */
```

```
}
```

Выборка по id

id - уникальный идентификатор элемента. Применяется как атрибут к любому тегу. Может быть использован на странице **только один раз!** CSS селектор начинается со знака “шарп” - #

html

```
<h1 id="unique">  
я уникальный  
</h1>
```

CSS

```
#unique {  
    /* стили */  
}
```

Выборка по классу

class - селектор который, в отличие от айди, может быть применен к любому количеству элементов. Благодаря этому происходит **наследование** стилей. CSS селектор начинается с точки

```
<p class="same-selector">  
    Текст параграфа.  
</p>
```

html

```
<a class="same-selector">  
    Ссылка, пытающаяся быть  
    параграфом  
</a>
```

```
.same-selectors {  
    font-size: 13px;  
    color: red;  
}
```

CSS

наложение стилей

html

```
<div class="first extra">
```

К этому блоку применены
стили обоих классов

```
</div>
```

CSS

```
.first {  
    color: red;  
}  
  
.extra {  
    font-size: 13px;  
}
```

перечисление селекторов

html

```
<a href="#">Я ссылка</a>  
<div class="element">  
  Просто блок  
</div>  
<p id="text">  
  Параграф текста  
</p>
```

CSS

```
a, .element, #text {  
  color: red;  
}
```

Синтаксис селекторов

parent и child

html

```
<div class="luke">
  Независимый люк!
</div>

<div id="vader">
  Люк я твой отец.

  <div class="luke">
    Так и знал что я твой сын!
  </div>
</div>
```

CSS

```
.luke {
  font-size: 13px;
  color: red;
}

#vader .luke {
  color: blue;
}
```

ЗАВИСИМОСТЬ селекторов

parent-child селектор так же называют относительным или селектором с зависимостью. Потому что отстилизированный таким образом элемент должен располагаться в HTML определенным образом.

```
<p class="paragraph">
  Lorem ipsum dolor sit <a href="" class="link">amet</a>.
</p>

<h2 class="another">
  Тут та же <a href="" class="link">ссылка</a>
</h2>
```

html

```
.paragraph .link {
  /*стили*/
}
```

css

составной селектор

Так же такой способ называют стилизацией при помощи **класса-модификатора**.

```
<div class="block">  
    Я просто элемент  
</div>  
<div class="block">  
    Я просто элемент  
</div>  
<div class="block red">  
    Я почти такой же, только в  
    красном цвете  
</div>
```

html

```
.block {  
    font-size: 15px;  
    color: red;  
}  
  
.block.red {  
    color: blue;  
}
```

css

ЗНАЧИМОСТЬ селекторов

Айди:

Значимость - 100

Класс:

Значимость - 10

Тег:

Значимость - 1

html

```
<a href="#" id="id" class="class">  
    Смотрим цвет?  
</a>
```

css

```
a {  
    color: red;  
}  
#id {  
    color: green;  
}  
.class {  
    color: black;  
}
```

Смотрим цвет?

браузер

!important – самый важный!

Конструкция “!important” - Повышает значимость селектора до максимума!

html

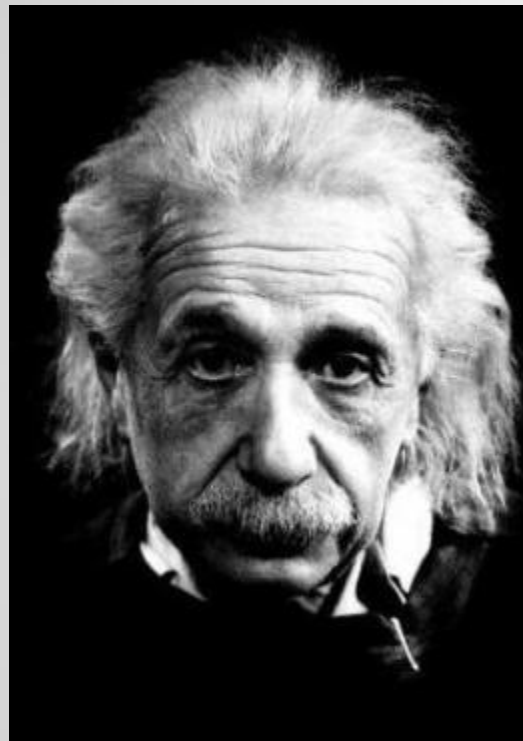
```
<a href="#" id="id" class="class">  
    Смотрим цвет?  
</a>
```

css

```
a {  
    color: green !important;  
}  
#id {  
    color: red;  
}  
.class {  
    color: black;  
}
```

Смотрим цвет?

браузер



«Ничего не понятно? Это нормально.»

- *Альберт Эйнштейн*