

# Введение в web- ТЕХНОЛОГИИ

Обзор web-технологий, выделение стека используемых нами технологий и области их применения

# Определение web-технологий

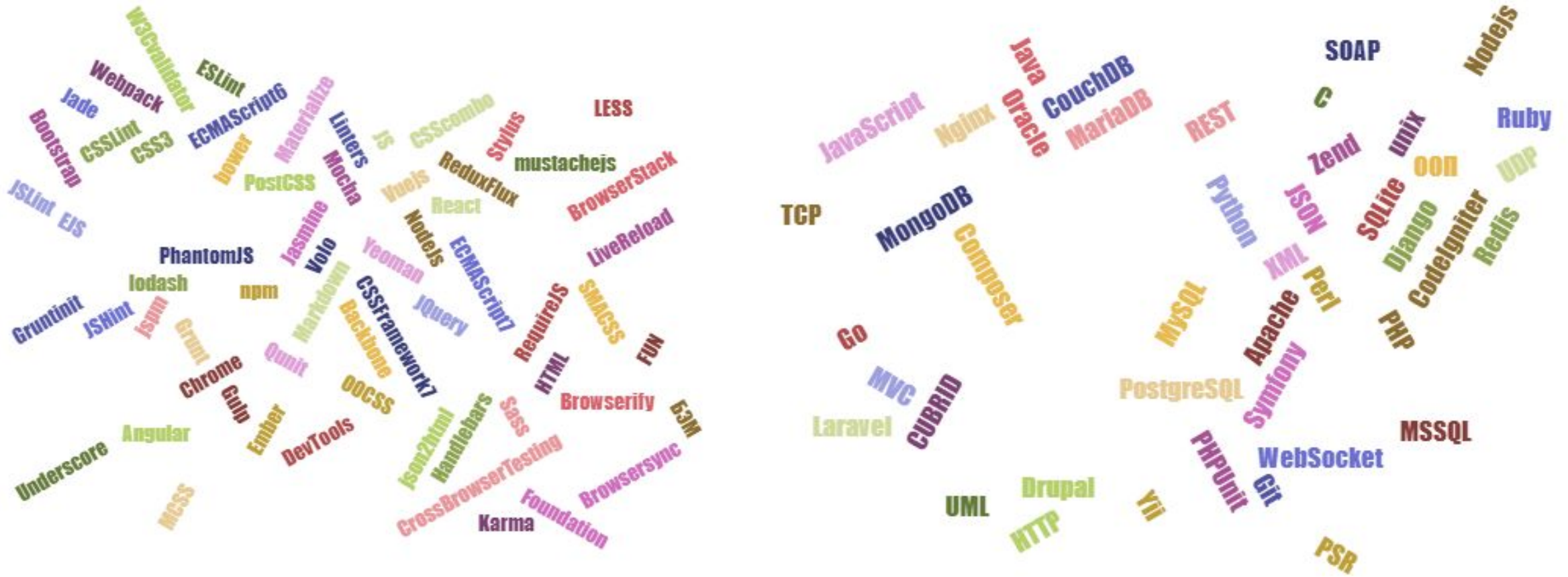
- **Web-технология** - набор программ для обеспечения взаимодействия клиент-сервер в сетях Internet или Intranet.
- **Сервер** (англ. *server* от *serve* — служить) в информационных технологиях — аппаратный или программный компонент вычислительной системы, выполняющий специализированные функции по запросу клиента, предоставляя ему доступ к определенным ресурсам. Сервер, реализованный в виде программы или программного модуля, обычно выполняет строго определённую задачу и обменивается информацией с клиентом по заранее определённому протоколу. Примерами программных серверов могут служить: файл-сервер, сервер печати, веб-сервер (Apache, IIS), сервер БД, X-сервер, почтовый сервер (Sendmail Postfix), OLE-сервер, ActiveX-сервер и т. п.
- **Клиент** — (в информационных технологиях) это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу. Программа, являющаяся клиентом, взаимодействует с сервером, используя определенный протокол. Она может запрашивать с сервера какие-либо данные, манипулировать данными непосредственно на сервере, запускать на сервере новые процессы и т. п. Полученные от сервера данные клиентская программа может предоставлять пользователю или использовать как-либо иначе, в зависимости от назначения программы. Программа-клиент и программа-сервер могут работать как на одном и том же компьютере, так и на разных.
- **Internet (интернет)** — всемирная система объединённых компьютерных сетей, построенная на использовании протокола IP и маршрутизации пакетов данных. Интернет образует глобальное информационное пространство.
- **Протокол** или **сетевой протокол** в компьютерных сетях — набор правил для специфического типа связи. Типы связи разбиты на уровни. Например, физический уровень (протоколы ISDN, RS-232), канальный уровень (протоколы Ethernet, Token ring, Fibre Channel), сетевой уровень (протоколы ICMP, IP, IPX), транспортный уровень (протоколы SPX, TCP, UDP, RTCP), прикладной уровень (протоколы binkp, DHCP, DNS, Finger, FTP, Gnutella, Gopher, HTTP, HTTPS, IMAP, IRC, Jabber, LDAP, NTP, NNTP, POP3, SSH, SMTP, Telnet).

# Классификация web-технологий

Все технологии, применяемые при создании веб-систем, делятся на два основных класса:

- **Frontend** - исполняемые на клиенте средствами обозревателя Интернет (HTML, CSS, JavaScript, Flash, ActiveX и т. п.) + вспомогательные технологии их разработки (LESS, SAAS, Gulp, WebPack и т.д.)
- **Backend** - исполняемые на сервере средствами веб-сервера (SSI, PHP, ASP, Perl, Python и т.п.) и связанных с ним систем (MySQL, PostgreSQL, MSSQL и т.п.)

# Наборы технологий frontend и backend



# Наш набор технологий

- PHP
- MySQL
- HTML
- CSS
- JS
- Git
- Composer
- MVC

# Применение этого стека технологий

с точки зрения возможностей

- Как пишут скромные разработчики языка: PHP может все (<http://php.net/manual/ru/intro-whatcando.php>)
- Изначально область применения языка сводилась к созданию не обремененных логикой и нагрузкой веб-сайтов
- На данный момент на нем разрабатывают backend для многофункциональных веб-приложений, социальных сетей, интернет-магазинов, форумов и т.д.
- С развитием ООП и оптимизацией, PHP познал что такое highload и биллинговые системы.
- Есть возможность использования php через командную строку

# Применение этого стека технологий

с точки зрения рынка

[https://kirov.hh.ru/search/vacancy?text=PHP&clusters=true&enable\\_snippets=true](https://kirov.hh.ru/search/vacancy?text=PHP&clusters=true&enable_snippets=true)

Итак PHP-программистов ищут:

- IT-фирмы, в т.ч. занимающиеся аутсорсингом
- Кредитные организации
- Службы доставки
- Ремонт квартир
- Торговые автоматы
- Книжные издательства
- И все остальные тоже...

# ПО для web-разработки

Мы будем использовать следующее ПО:

- Web-сервера (nginx, apache)
- IDE (PhpStorm, NetBeans)
- MySQL клиент (navicat, mysql workbench, phpMyAdmin)
- Менеджер зависимостей (composer)
- Системы контроля версий (git, svn)



# Web-сервера

Web-сервер это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потокком или другими данными.

Web-сервером называют как программное обеспечение, выполняющее функции web-сервера, так и непосредственно компьютер, на котором это ПО работает. Варианты ПО:

- Apache (<https://httpd.apache.org/>)
- Nginx (<https://nginx.ru/ru/>)
- Lighttpd (<https://www.lighttpd.net/>)

# Наша конфигурация web-сервера

- Мы используем связку nginx + apache (<https://habrahabr.ru/post/120814/>)
- Альтернатива - nginx + php-fpm (<https://habrahabr.ru/post/320036/>)
- Для разработки я использую Windows + WAMP (<http://www.wampserver.com/ru/>)
- Альтернатива - OpenServer (<https://ospanel.io/>) для Windows и встроенный веб-сервер для Linux

# IDE (*Integrated development environment*)

- Мы используем PhpStorm (<https://www.jetbrains.com/phpstorm/>)
- Альтернатива – NetBeans (<https://netbeans.org/>)
- О других IDE вкратце сказано в статье [https://geekbrains.ru/posts/php\\_ides](https://geekbrains.ru/posts/php_ides)
- Преимущества IDE перед простыми редакторами <https://habrahabr.ru/post/265197/>
- Полезные фишки PhpStorm <https://habrahabr.ru/post/212077/>  
<https://habrahabr.ru/post/157409/>

# Другое ПО

- MySQL клиент (navicat - <https://www2.navicat.com/ru/>, mysql workbench - <https://www.mysql.com/products/workbench/>, phpMyAdmin - <https://www.phpmyadmin.net/>)
- Менеджер зависимостей (composer - <https://getcomposer.org/>)
- Системы контроля версий (git - <https://git-scm.com/>, svn - <https://subversion.apache.org/>)

# Использованная ~~ссылки~~ литература

- [http://math.csu.ru/~rusear/W\\_T/Intredution/Int.htm](http://math.csu.ru/~rusear/W_T/Intredution/Int.htm)
- <https://studfiles.net/preview/6008269/page:3/>
- <http://php.net/manual/ru/>
- <https://kirov.hh.ru/>
- <https://habrahabr.ru/>

# Практическая часть

На этом и следующем занятии будем изучать основы PHP, задания будут не только к этому но и подготовкой к следующему занятию.

Области, которые мы затронем:

- Основы синтаксиса, типизация
- Управляющие конструкции (условные операторы, циклы)
- Работа с файлами
- Сессии
- Рекурсия
- Статические переменные

Массивы

# Задание 1, напишите программу, выводящую 3 раза «Hello, World!» в следующих вариантах:

- одним оператором  
(<http://php.net/manual/ru/language.basic-syntax.php>)
- Процедурным кодом  
(<http://php.net/manual/ru/language.operators.php>)
- Через цикл (<http://php.net/manual/ru/language.control-structures.php>)
- Через массив (<http://php.net/manual/ru/book.array.php>)
- Через строковые функции (<http://php.net/manual/ru/ref.strings.php>)
- С использованием функции  
(<http://php.net/manual/ru/language.functions.php>)
- В парадигме ООП (<http://php.net/manual/ru/language.oop5.php>)
- \*С использованием паттерна MVC
- \* Сравните все указанные выше методы, укажите их + и -

# Задание 2, типизация, условные операторы

<https://returntrue.win/>



# Задание 3, напишите функцию, которая считает количество пятниц в году, входной параметр - ГОД

- Просто функцию
- Класс со статическим методом  
(<http://php.net/manual/ru/language.oop5.static.php>)
- С кешированием в сессию  
(<http://php.net/manual/ru/session.examples.basic.php>)
- С кешированием в статическую переменную  
(<http://php.net/manual/ru/language.variables.scope.php>)
- С кешированием в файл  
(<http://php.net/manual/ru/book.filesystem.php>)
- \* С кешированием в memcached  
(<http://php.net/manual/ru/book.memcached.php>)
- \* С кешированием в разные источники из предыдущих в зависимости от их доступности

# Задание 4, напишите функцию, которая выполняет объединение произвольных многомерных массивов

```
$ar1 = [  
  'String1',  
  'Key1' => 'String2',  
  'Key2' => [  
    'Subkey1' => 'String3',  
    [  
      1000,  
      'Subsubkey1' => 10000,  
      'Subsubkey2' => -1,  
      10 => [  
        1,  
        2,  
        3,  
      ],  
    ],  
  'Subkey2' => 'String4',  
];
```

```
$ar2 = [  
  'String22',  
  'Key11' => 'String222',  
  'Key2' => [  
    'Subkey111' => 'String33',  
    [  
      100,  
      'Subsubkey1111' => 100,  
      'Subsubkey2' => -100,  
      100 => [  
        5,  
        6,  
        7,  
      ],  
    ],  
];
```

# Задание 5, напишите функцию, считающую количество файлов конфигурации в каталоге

- Структура каталога и подкаталогов должна быть произвольной
- Файл конфига определять по имени – оно должно содержать слово «config» в любом регистре
- Исключить заведомо левые расширения, например рисунки и видео
- Исходную папку для анализа можно задавать прямо в коде
- Вывести на экран все файлы конфигурации с прямыми путями к ним, их размером и датой последней модификации
- \* Повторяющиеся файлы переместить в отдельную директорию