

Программно-аппаратные средства Web

**Веб-серверы. CGI, модули,
FastCGI, WSGI.**

Сергей Геннадьевич Синица

КубГУ, 2021

sin@kubsu.ru

Apache

Раздавать статические ресурсы

Uri = имя файла относительно DocumentRoot

Раздавать динамические ресурсы

CGI – Common Gateway Interface

Запуск на сервере практически произвольной программы с параметрами

Модули – расширения apache в виде специальных .so файлов

Например, интерпретаторы php, perl, python, модуль WSGI

FastCGI – протокол для общения со специальным сервером для запуска скриптов

fastcgiwrapper

PHP FPM – менеджер процессов FastCGI PHP

uWSGI — менеджер процессов Python

Функционал

Виртуальные директории

Виртуальные хосты

Mod_rewrite

Proxy redirect

Режим запуска (sudo a2query -M)

Prefork

Worker

Event

NGINX

Аналогичен Apache, не поддерживает модули.

Эффективная Event-модель обработки соединений.

Высокопроизводительный веб-сервер.

Динамические страницы отдаются через FastCGI или проксированием на сервер-приложений.

Node.JS

Динамическая нестрогая типизация, но есть TypeScript!

Встроенный сервер HTTP, используется с прокси NGINX и менеджером процессов, например PM2.

Асинхронная однопоточная обработка.

Высокая производительность.

```
var http = require("http");
```

```
http.createServer(function(request, response) {  
  response.writeHead(200, {"Content-Type": "text/plain"});  
  response.write("Hello World");  
  response.end();  
}).listen(8888);
```

PHP

Динамическая нестрогая типизация, но есть тайп хинтинг!

Встроенный сервер HTTP (ReactPHP), но обычно используется через PHP FPM или модуль Apache.

Однопоточная обработка.

Высокая производительность.

```
<?php  
print("Hello world");
```


Python

Динамическая строгая типизация.

Встроенный сервер HTTP есть (Tornado), но используется обычно в режиме WSGI.

Многопоточная обработка, но GIL мешает.

Низкая производительность.

```
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
```

```
class HttpProcessor(BaseHTTPRequestHandler):
```

```
    def do_GET(self):
```

```
        self.send_response(200)
```

```
        self.send_header('content-type','text/html')
```

```
        self.end_headers()
```

```
        self.wfile.write("hello !")
```

```
serv = HTTPServer(("localhost", 8888), HttpProcessor)
```

```
serv.serve_forever()
```

Go

Статическая строгая типизация.

Встроенный сервер HTTP.

Многопоточная обработка.

Высокая производительность.

```
package main
```

```
import (  
    "fmt"  
    "net/http"  
)
```

```
func main() {  
    http.HandleFunc("/", func (w http.ResponseWriter, r *http.Request) {  
        fmt.Fprintf(w, "Welcome to my website!")  
    })
```

```
    fs := http.FileServer(http.Dir("static/"))  
    http.Handle("/static/", http.StripPrefix("/static/", fs))
```

```
    http.ListenAndServe(":80", nil)  
}
```

Сравнение языков Web-разработки

Python 1991
PHP 1995
Node.JS 2009
Java 1995
C# 2000
Go 2009

Типизация
Популярность языка
Скорость разработки и деплоя
Производительность
Стандарты
Обратная совместимость
Масштабируемость

MySQL / MariaDB

Запуск клиента:

```
mysql -uuser -ppassword database
```

Создать таблицу:

```
CREATE TABLE application (  
  id int(10) unsigned NOT NULL AUTO_INCREMENT,  
  name varchar(128) NOT NULL DEFAULT "",  
  PRIMARY KEY (id)  
);
```

Вставить запись:

```
INSERT INTO application SET name = 'name';
```

PHP / PDO

// Сохранение в базу данных.

```
$user = 'db';  
$pass = '123';  
$db = new PDO('mysql:host=localhost;dbname=test',  
    $user, $pass,  
    [PDO::ATTR_PERSISTENT => true]);  
  
// Подготовленный запрос. Не именованные метки.  
try {  
    $stmt = $db->prepare("INSERT INTO application (name) SET name = ?");  
    $stmt -> execute(['fio']);  
}  
catch(PDOException $e){  
    print('Error : ' . $e->getMessage());  
    exit();  
}
```

PHP

// Именованные метки.

```
$stmt = $db->prepare("INSERT INTO test (label,color) VALUES (:label,:color)");
```

```
$stmt -> execute(['label'=>'perfect', 'color'=>'green']);
```

// Еще вариант

```
$stmt = $db->prepare("INSERT INTO users (firstname, lastname, email) VALUES (:firstname, :lastname, :email)");
```

```
$stmt->bindParam(':firstname', $firstname);
```

```
$stmt->bindParam(':lastname', $lastname);
```

```
$stmt->bindParam(':email', $email);
```

```
$firstname = "John";
```

```
$lastname = "Smith";
```

```
$email = "john@test.com";
```

```
$stmt->execute();
```

MySQL

На Node JS используем модуль `mysql2`.

На Python используем пакет `mysql.connector`

Использование подготовленных запросов обязательно!

Ссылки

<https://www.php.net/docs.php>

<http://www.nodebeginner.ru/>

<https://dev.mysql.com/doc/>