



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

# Какво научихме дотук

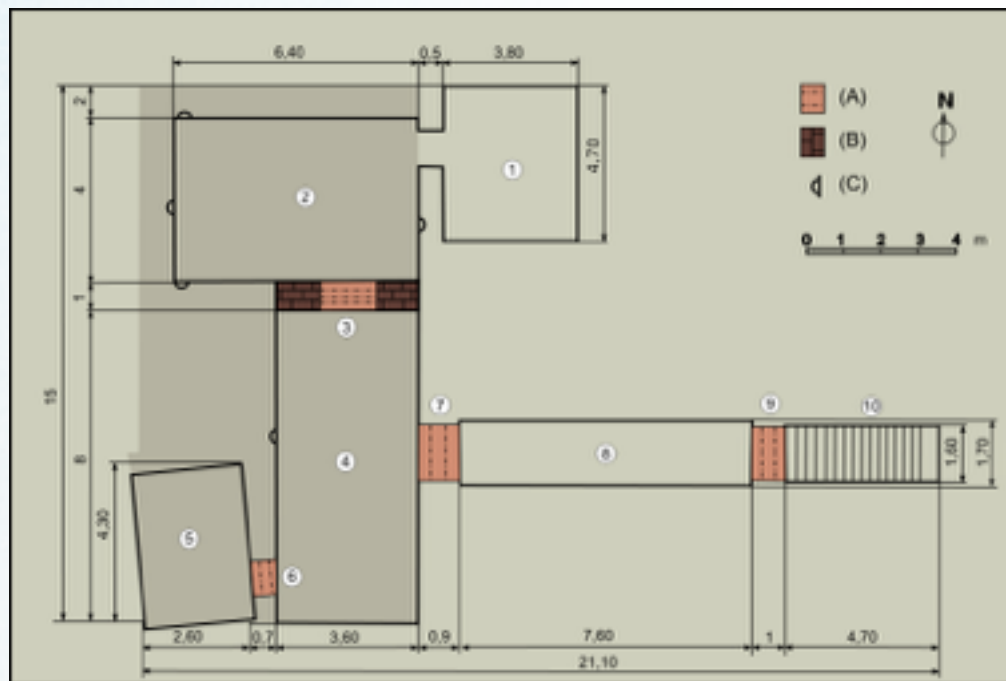
- Как да съставяме HTML страници
  - скелет на html страница
  - основните html елементи
- Как да стилизираме HTML текст и съдържание
  - как да украсяваме html съдържанието
  - как да подреждаме и позиционираме html съдържанието
  - как да пишем responsive страници



# Какво научихме дотук

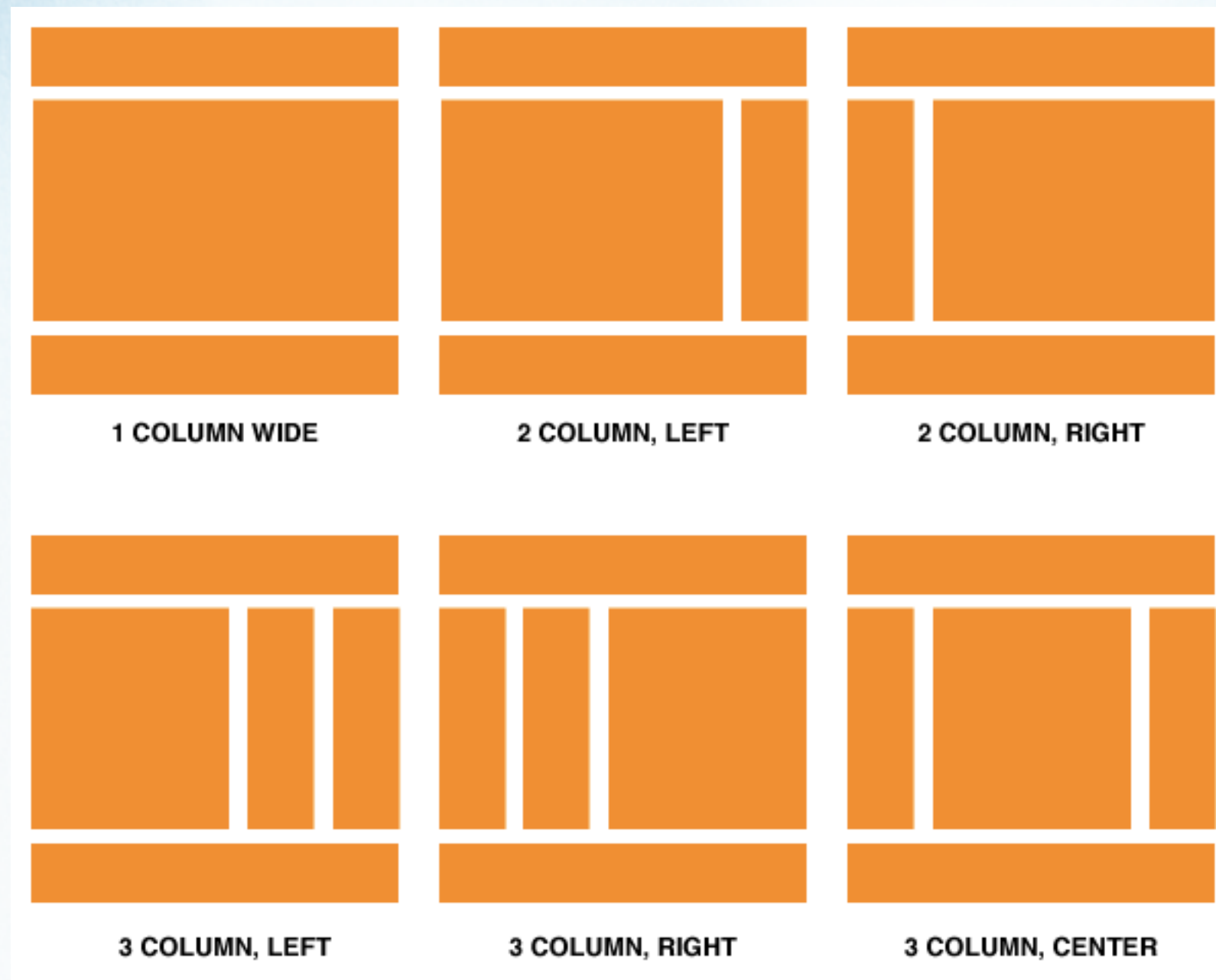
- Навигация
  - КЪМ ВЪНШНИ САЙТОВЕ
    - `<a href="http://.." target="_blank">`
  - КЪМ ВЪТРЕШНИ ПРЕПРАТКИ
    - `<a href="#my-section">`
    - `<a href="contact.html">`
- Layout
- Frameworks

# КАКВО ВСЪЩНОСТ БЕШЕ layout?



това е layout





т.е. **layout**-а -това е наместването на разни правоъгълници по екрана



# какво точно беше framework?

правенето на уебсайт със и без помощта на framework, може да се сравни с ..



това да трябва да приготвим сложно ястие със и без подходящи инструменти и условия



т.е. **framework**-а е един сложен и мощен инструмент (или набор от инструменти), с който ако веднъж се научим да боравим, ще започне да ни пести наистина много време и усилия



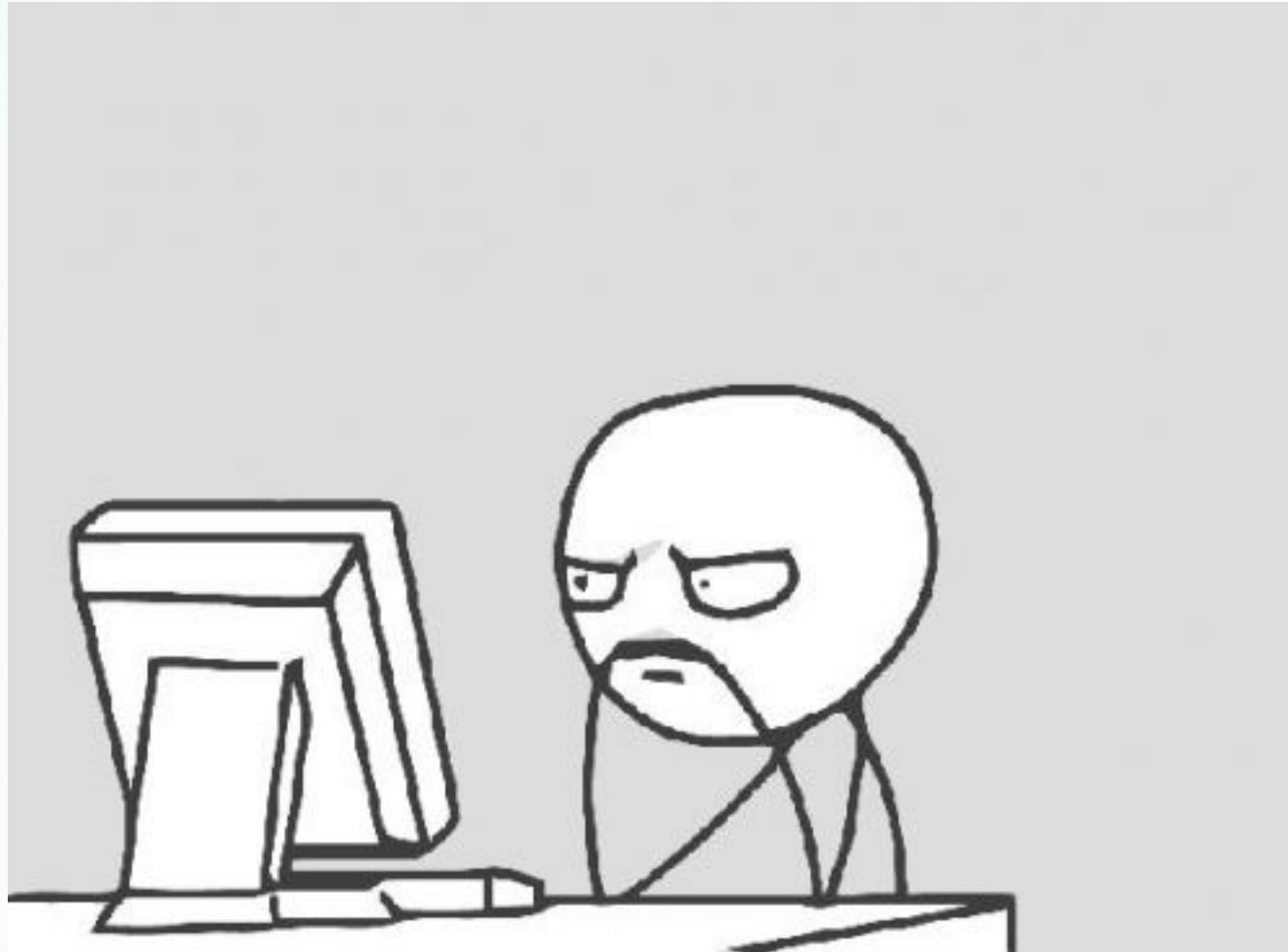


Научихме също какво е конвенция и styleguides.

Вече знаем как да пишем хубав, подреден и разбираем код. Изобщо - как да бъдем любезни и отговорни фронт-енд програмисти :)



OK, I did exactly as they said..,



and it doesn't work

## 2 съвета:

### 1. Търпение и практика

- никой не е научен или може а се научи само от един туториал
- “Насила хубост не става”

### 2. Всеки проблем има решение, а повечето от тях са в <http://stackoverflow.com/>



# Примера от лекция 2

- Направихме aside, но той стои “тъпо” - не иска да се подравни вертикално с body и main
- “Over two decades of intense development of the HTML technology and still there’s no straight forward way to vertically align things” — Evan Czaplicki (creator of Elm)

Are you flexible enough?



# FLEXBOX

- [http://www.w3schools.com/css/css3\\_flexbox.asp](http://www.w3schools.com/css/css3_flexbox.asp)
- <http://stackoverflow.com/questions/20547819/vertical-align-with-bootstrap-3#answer-25517025>
- <http://html5please.com/#flexbox>

## 3-те решения на домашно 2:

- <http://zenifytheweb.com/courses/lessons/lesson2/example/aside-solution-inline-block/index.html>
- <http://zenifytheweb.com/courses/lessons/lesson2/example/aside-solution-display-table/index.html>
- <http://zenifytheweb.com/courses/lessons/lesson2/example/aside-solution-flex/index.html>



# Последното домашно

- <http://zenifytheweb.com/courses/lessons/lesson7/example/index.html>
- media queries
- vertical align
- add a new breakpoint

# HTML head

- [http://www.w3schools.com/tags/att\\_link\\_rel.asp](http://www.w3schools.com/tags/att_link_rel.asp)
- [http://www.w3schools.com/tags/tag\\_meta.asp](http://www.w3schools.com/tags/tag_meta.asp)
- Още:
  - <http://ogp.me/>
  - <http://schema.org/>

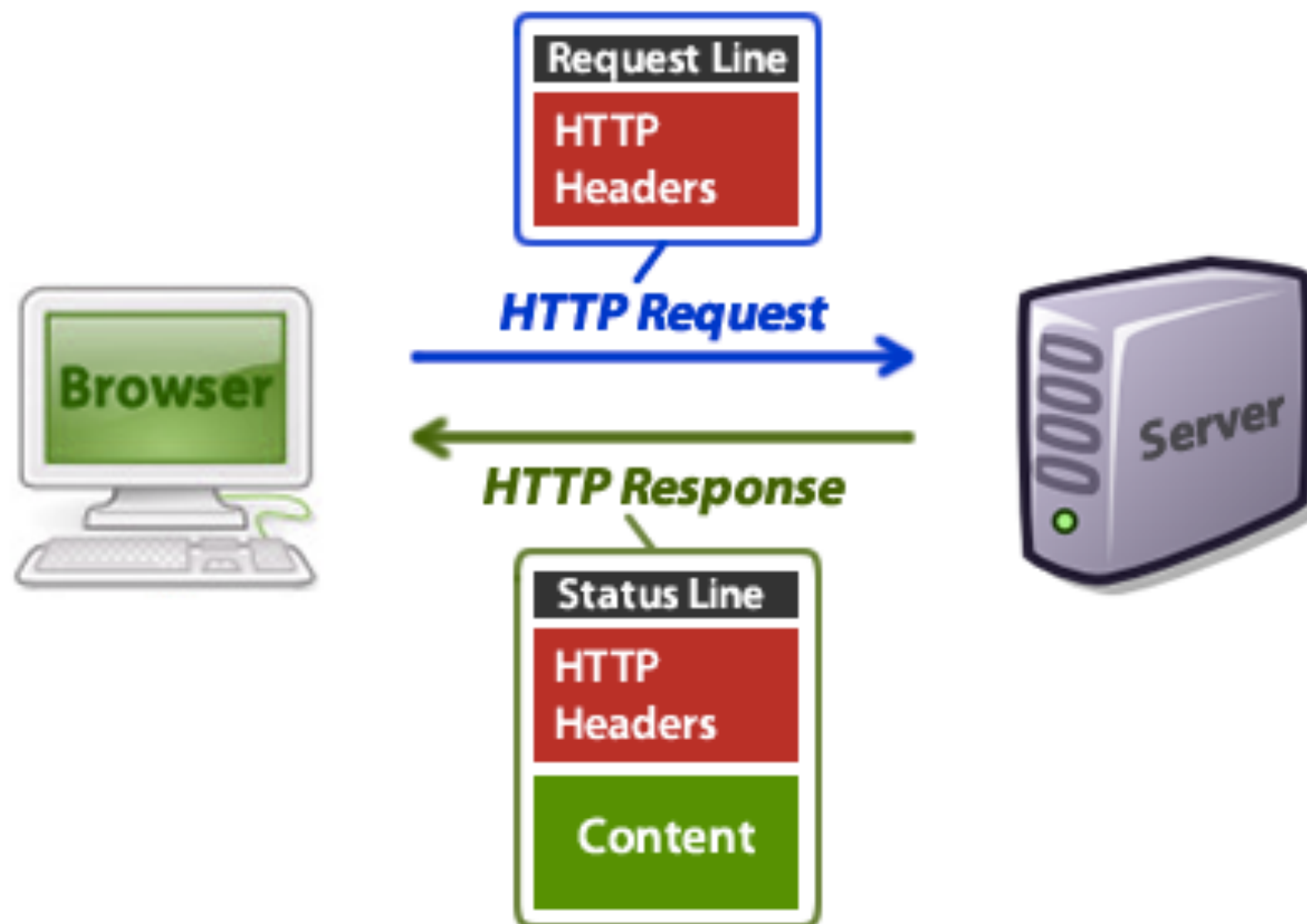


# HTML5 Global data

- Използваме ги за да съхраняваме допълнителна информация за даден елемент, за която няма валиден html атрибут
- Например
  - за да си запазим “state” (някакво междинно състояние), когато боравим с front-end логика (javascript)
  - при използването на generators като angular (data-ng-\*)
- data-\* е валиден таг!
- [http://www.w3schools.com/tags/att\\_global\\_data.asp](http://www.w3schools.com/tags/att_global_data.asp)

# HTTP





- Hyper Text Transfer Protocol
- Това е протокол (правилник) за това как клиента и сървъра обменят информация
- Клиента (браузъра) прави request към сървъра, като му изпраща съобщение (message), което съдържа:
  - request line
  - http headers
  - body (само при POST заявки)



Request line-a се състои от **име на метод, път и протокол** (име и версия):

method	path	protocol
GET	/tutorials/other/top-20-mysql-best-practices/	HTTP/1.1

```
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

**HTTP headers as Name: Value**

- Когато request-а стигне до сървъра, сървъра го “прочита” и намира документа, който е поискан в request-а
- За да изпрати документа към клиента, сървъра съставя т.нар response message (отговор), който се състои от:
  - status line
  - http headers
  - content (самия документ, например съдържанието на index.html)



```
HTTP/1.x 200 OK
Transfer-Encoding: chunked
Date: Sat, 28 Nov 2009 04:36:25 GMT
Server: LiteSpeed
Connection: close
X-Powered-By: W3 Total Cache/0.8
Pragma: public
Expires: Sat, 28 Nov 2009 05:36:25 GMT
Etag: "pub1259380237;gz"
Cache-Control: max-age=3600, public
Content-Type: text/html; charset=UTF-8
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
X-Pingback: http://net.tutsplus.com/xmlrpc.php
Content-Encoding: gzip
Vary: Accept-Encoding, Cookie, User-Agent

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8
<title>Top 20+ MySQL Best Practices - Nettuts+</title>
<!-- ... rest of the html ... -->
```

Винаги първият ред от съобщението отговор съдържа т.нар. **status line**, който от своя страна се състои от:

- протокол (име и версия)
- status код

**protocol**      **status code**

**HTTP/1.x** **200 OK**

```
Transfer-Encoding: chunked
Date: Sat, 28 Nov 2009 04:36:25 GMT
Server: LiteSpeed
Connection: close
X-Powered-By: W3 Total Cache/0.8
Pragma: public
Expires: Sat, 28 Nov 2009 05:36:25 GMT
Etag: "pub1259380237;gz"
Cache-Control: max-age=3600, public
Content-Type: text/html; charset=UTF-8
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
X-Pingback: http://net.tutsplus.com/xmlrpc.php
Content-Encoding: gzip
Vary: Accept-Encoding, Cookie, User-Agent
```

**HTTP headers as Name: Value**



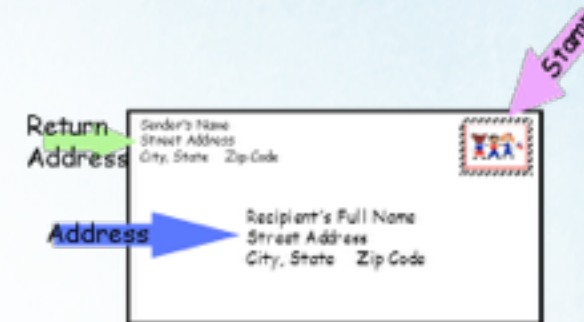
- От върнатият status code, клиента (браузърът) може да разбере дали requesta (заявката, която е направил) е бил успешен или не
  - 200 - OK
  - 404 - Not found
  - 403 - Forbidden
  - <http://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039>

- Представете си че съобщенията, които си изпращат клиента и сървъра са пощенски телеграми и писма
- клиента изпраща на сървъра една телеграма (или картичка), където:
  - получателя е request line-a (кой трябва да получи съобщението)
  - данните на подателя са http headers
  - марката представлява името и версията на протокола (пак в request line-a)
- Самият протокол - това са правилата за това как се пише подател и получател. Когато са изписани правилно, съобщението ще бъде доставено успешно





- На телеграмата, сървъра отговаря с писмо
- Не е нужно да пише адрес на получател, защото го връща заедно с обратната разписка (клиента търси сървъра и го пита, а сървъра само отговаря. Сървъра НЕ търси клиента)
- На плика най-отгоре записва дали картичката му е харесала или не, а отдолу разни пояснения към условията, в които е приел заявката (http headers)
  - Ако му е харесала (200 OK), в плика има и писмо със съдържанието на изкания документ
  - Ако пък не му е харесала - плика е празен :)
- Все пак и той слага марка - името и версията на протокола



# Въпроси?



# Домашно

<https://github.com/zzeni/swift-academy-homeworks/tree/master/tasks/L8>

# Примери

<http://zenifytheweb.com/courses/lessons/lesson8/example/index.html>