

# Увод в уеб технологии и дизайн

Уеб браузъри, сървъри, HTML, CSS и други опасни думички  
:)

Me : I just started  
"programming" in HTML

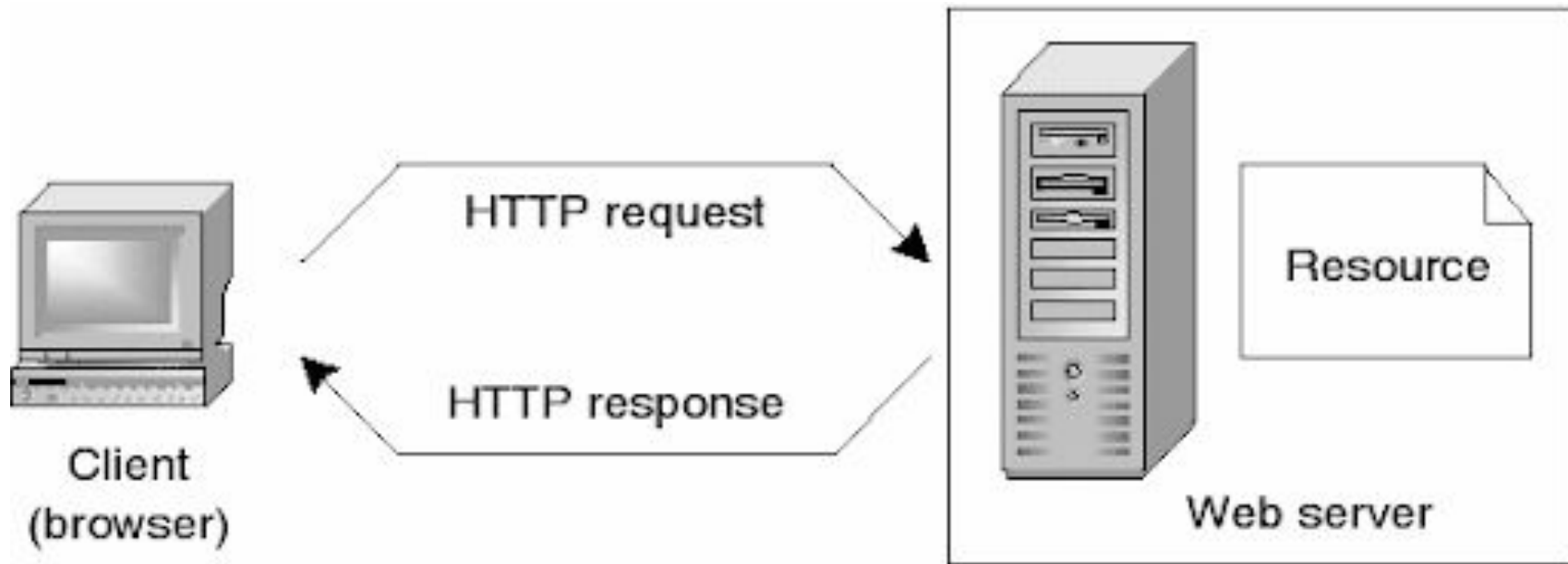


# Малко за HTTP

- HyperText Transfer Protocol - протокол за пренос на хипертекст
- Позволява публикуването на HTML страници
- Използва се от уеб браузърите (наричат се още клиенти) и уеб сървърите
- Уеб сървърът е приложна програма, която позволява на дадена машина да предоставя информация за други машини под формата на хипертекст

# Как работи HTTP?

Клиентът изпраща заявка (request) за достъп до даден ресурс.  
Сървърът връща заявения ресурс като отговор (response).



# HTTP Заявка

- HTTP заявката се състои от следните елементи:
  - Метод на заявката - описва действието, което трябва да се извърши от заявката
  - Цел (назначение) на заявката - обикновено това е URL адреса
  - Версия (HTTP протоколът има различни версии, актуалната е HTTP/2.0, като се подготвя и 3.0)
  - Служебна информация (Headers)
  - Тяло на заявката (Request body)

# HTTP Заявка

```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request  
Message  
Header

A blank line separates header & body

Request Message Body

# HTTP Отговор

Частите на HTTP отговора са:

- Версия на протокола
- Статус на отговора, например: 200, 404, 302, 500... (проучете в Интернет различните видове статуси)
- Текст, описващ статуса
- Служебна информация (headers)
- Тяло (body) <- Тук често се съдържа HTML кода :)

# Видове (методи) HTTP заявки

- GET - извличане на ресурс
- POST - изпращане на информация към сървъра
- PUT - обновява/замества ресурс
- DELETE - изтриване на ресурс

Допълнително информация:

<https://www.restapitutorial.com/lessons/httpmethods.html>



# HTML

- HTML – HyperText Markup Language
- Описателен (Мета) език
- Описва какви „обекти” трябва да има в един уеб документ
- Грозен и обикновен без CSS и JavaScript
- Работи с тагове (<p></p>, <div></div>, 
- Таговете си имат атрибути (например src на img)
- Определени тагове се затварят и между тях се поставя съдържание и/или други тагове, но има и такива, които не се затварят
- Къде да го научим: <https://www.w3schools.com/>

# CSS

- CSS - Cascading StyleSheets
- Обектите създадени чрез HTML си имат свойства. CSS помага за именянето на тези свойства
- CSS работи с правила, които задават свойства за определен набор от елементи.
- Елементите се избират чрез селектори, които отразяват техния вид и положение в йерархията (йерархия - един елемент може да се намира „вътре” в друг елемент - тогава казваме, че вътрешния е **дете** на външния, а външния е **родител**)

# JavaScript

- За разлика от другите две JavaScript е пълноценен програмен език
- Основното му предназначение е да манипулира DOM
- DOM - Document Object Model - съдържа в себе си всички елементи, атрибути и свойства, които са създадени в браузъра
- JS може да добавя нови елементи, да изтрива елементи или пък да променя съдържанието, атрибутите или стиловите свойства на вече съществуващи елементи
- JS притежава редица други функционалности
- JS, като част от сайта, в общия случай се изпълнява от интерпретатор в уеб браузъра
- JS е подобен на C# от синтактична гледна точка

# Uniform Resource Locator (URL)

`http://mysite.com:8080/demo/index.php?id=27&lang=en#lectures`



Протокол  
фрагмент



хост



порт



път



низ на заявката



# URL

URL адресът е форматиран низ, състоящ се от:

- Протокол за комуникация (http, ftp, https...) – HTTP в повечето случаи
- Хост или IP адрес (www.mon.bg, gmail.com, 127.0.0.1, web)
- Порт (стандартният порт е 80) – число в обхвата [0...65535]
- Път (/forum, /path/index.php)
- Низ за заявка(?id=27&lang=en)
- Фрагмент (#lectures) – използва се от браузъра за навигация към някоя секция

# Низ за заявка

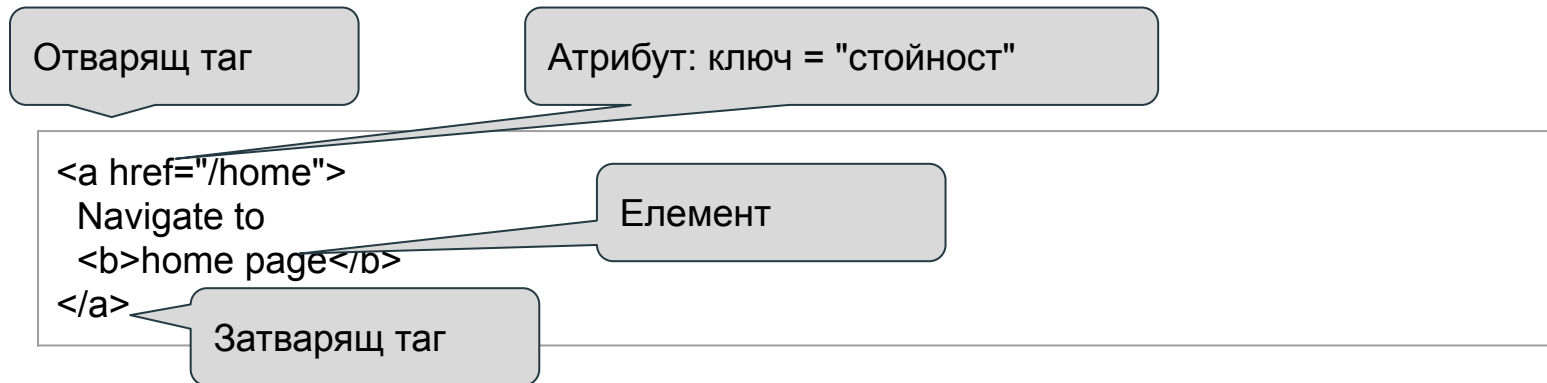
- Съдържа данни, които не са част от структурата на пътя
- Често използван при търсения и динамични страници
- Частта от URL след въпросителен знак (?)
- Множество параметри са разделени от разделители

# HTML терминология

Тагове – формират елементите в HTML

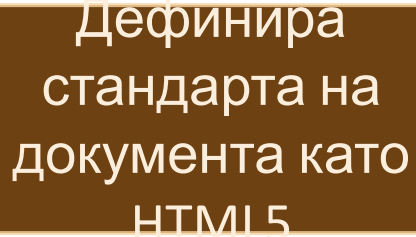
Атрибути – свойствата на таговете - размер, цвят и т.н

Елементи – комбинация от отварящ, затварящ таг и атрибути



# Вашата първа HTML страница – Пример

**<!DOCTYPE html>**



Дефинира  
стандарта на  
документа като  
HTML5



# Вашата първа HTML страница – Пример

`<!DOCTYPE html>`

`<html>`

Основният елемент  
на една HTML  
страница

`</html>`

Забележка: Повечето  
HTML тагове трябва да  
бъдат затваряни

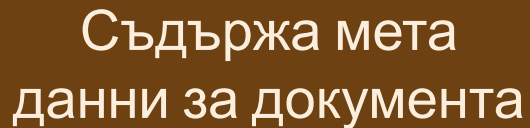
# Вашата първа HTML страница – Пример

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
  </head>
```



Съдържа мета  
данни за документа

```
</html>
```

# Вашата първа HTML страница – Пример

```
<!DOCTYPE html>
```

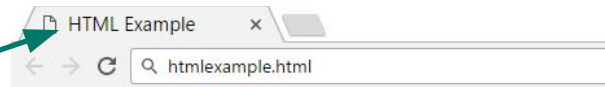
```
<html>
```

```
  <head>
```

```
    <title>HTML Example</title>
```

```
  </head>
```

```
</html>
```



Определя  
заглавието на  
документа в  
браузъра

# Вашата първа HTML страница – Пример

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

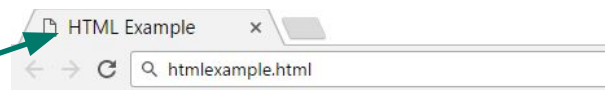
```
    <title>HTML Example</title>
```

```
  </head>
```

```
  <body>
```

```
    </body>
```

```
</html>
```



Съдържа видимото  
съдържание на  
страницата

# Вашата първа HTML страница – Пример

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML Example</title>
```

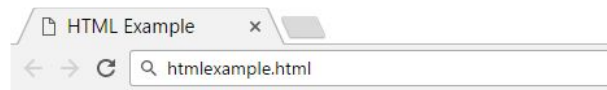
```
  </head>
```

```
  <body>
```

```
    <h1>Hello HTML!</h1>
```

```
  </body>
```

```
</html>
```

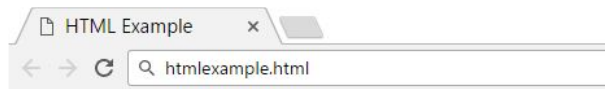


**Hello HTML!**

**Дефинира голямо  
заглавие**

# Вашата първа HTML страница – Пример

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Example</title>
  </head>
  <body>
    <h1>Hello HTML!</h1>
    <p>HTML describes formatted text using
tags.</p>
  </body>
</html>
```



**Hello HTML!**

HTML describes formatted text using tags.

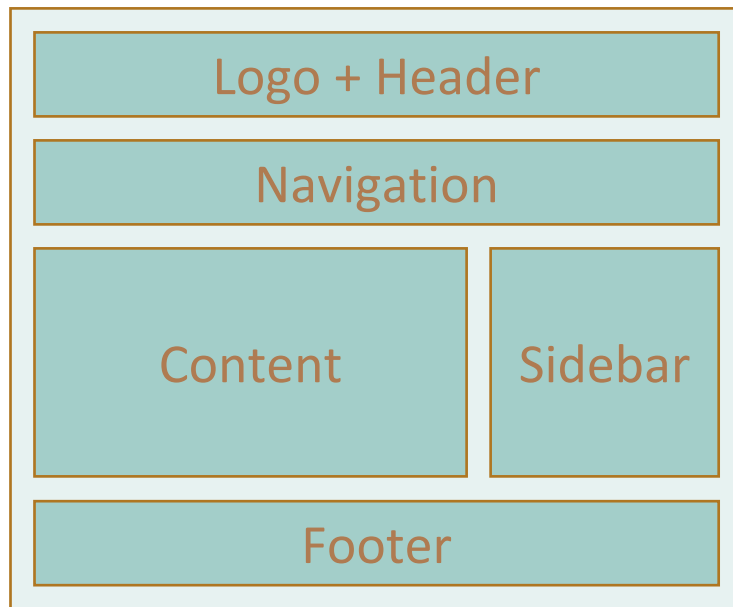
**Дефинира  
параграф**

# Семантични тагове в HTML5

**В HTML5 има семантични (значеши) тагове за оформление <header>, <footer>, <nav>, <aside>, <section>, <main> и др. Тези тагове помагат на търсачките (и браузърите) да обработват по-добре сайта ви.**

# Семантични тагове в HTML5

```
<html>
  <head> ... </head>
  <body>
    <header> ... </header>
    <nav> ... </nav>
    <aside> ... </aside>
    <section> ... </section>
    <footer> ... </footer>
  </body>
</html>
```





# Благодаря за вниманието!

Автор: Петър Р. Петров, учител по програмиране, ПГЕЕ  
„Константин Фотинов“, гр. Бургас