



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

# HTML attributes



- Вече знаем, че **html** елементите могат да имат атрибути, които се задават по следния начин:  
`<име-на-елемент attribute1="value1" attribute2="value2" ... >`
- т.е. във формат: **ключ="стойност"**
- примери за често използвани атрибути: **id, src, href, alt, title ...**
- за атрибутите важат следните правила:
  - винаги трябва да ползваме кавички за стойността на атрибута
  - винаги трябва да изписваме ключа с малки букви
  - атрибутите могат да се слагат **единствено** в *отварящия таг* на елемента

# Глобални атрибути

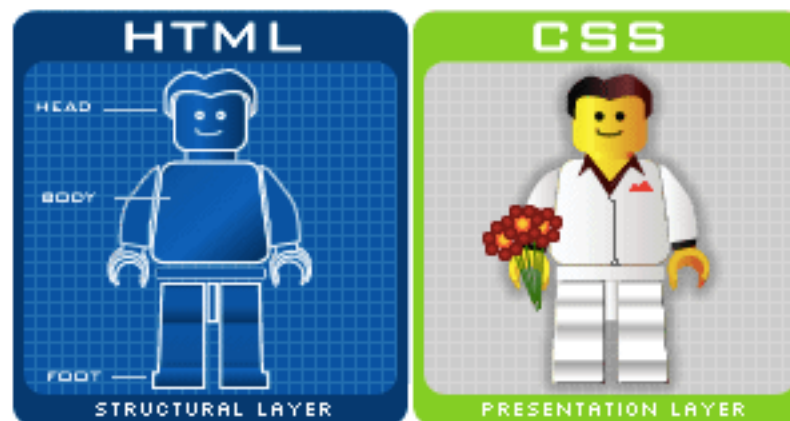
- Повечето от `html` атрибутите са специфични за конкретни *html* елементи, като например **href** на линковете, **src** на картинките и т.н.
- Има обаче набор от атрибути, които могат да се ползват от **абсолютно** всички *html* елементи и те се наричат **глобални**
  - глобални атрибути са: **id**, **class**, **style**, **title** и т.н.
  - специфични атрибути:
    - **src**: използва се от **img**, **video**, **audio**, **iframe**, **script** и т.н.
    - **href**: използва се от **a** и **link**
    - и още много други: **type**, **alt**, **width**, **height**, **name**, **value** и т.н.

[https://developer.mozilla.org/en-US/docs/Web/HTML/Global\\_attributes](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes)

[http://www.w3schools.com/tags/ref\\_standardattributes.asp](http://www.w3schools.com/tags/ref_standardattributes.asp)



# CSS



# Как се ползва?

За да включим CSS стилизиране в нашата HTML страница, използваме един от следните подходи:

- с линк (препоръчително):

```
<link rel="stylesheet" href="css/style.css">
```

- директно в HTML-а чрез **style** тагове (трябва да са в head частта):

```
<style>
```

```
  body {
```

```
    font-family: Helvetica;
```

```
  }
```

```
</style>
```

- или просто inline като **style** атрибут на елемента: **<body style="font-size: 25px">**

# Синтаксис

```
selector1 {  
  property1: value1;  
  property2: value2;  
  ...  
}
```

```
selector2 {  
  property1: value1;  
  property2: value2;  
  ...  
}
```

```
/* това е коментар и не се чете от браузъра */
```



# Правила

Конструкцията:

```
selector {  
  property: value;  
  ...  
}
```

се нарича правило.

Т.е. CSS файла представлява набор от правила, които задават стилизирането на различните HTML елементи

Селекторите указват за кои елементи важи правилото, а тялото на правилото (отделните properties) - какви конкретни стилове ще се приложат на съответните елементи



# Селектори

- Селектора ни дава възможност да укажем за кои елементи важат стиловете от правилото
- Например, имаме следната структура:

```
<body>  
  <section id="main-section">  
    <h1>My main section</h1>  
  </section>  
</body>
```
- Както вече знаем, за да зададем стил на `body` елемента, използваме `body{...}` записа, за стил на хединга - `h1{...}` записа.
- Т.е. **body**, **section** и **h1** са селектори

- Други селектори за същия код са
  - `body section { ... }`
  - `body h1 { ... }`
  - `body section h1 { ... }`
  - `section h1 { ... }`
  - `#main-section { ... }`
  - и дори: `#main-section h1 { ... }`
- Т.е. използването на id атрибута, ни дава селектор за този елемент
- [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)



# HTML

## **id** и **class** на елемента

- *id* и *class* са атрибути, които могат да ползват от *всеки* html element
- ***id*** атрибута задава уникален идентификатор на съответния елемент. Той трябва да е винаги различен за различните елементи

```

```

- ***class*** атрибута задава наименование на клас, към който html елемента принадлежи. Използва се за групиране на сходни елементи и идентифицира уникално група от елементи

```

```



# id и class в CSS-a

- HTML:

```
<p id="unique">First</p>  
<p class="grouped">Second</p>  
<p class="grouped">Third</p>  
<p>Just a regular p without id or class</p>
```

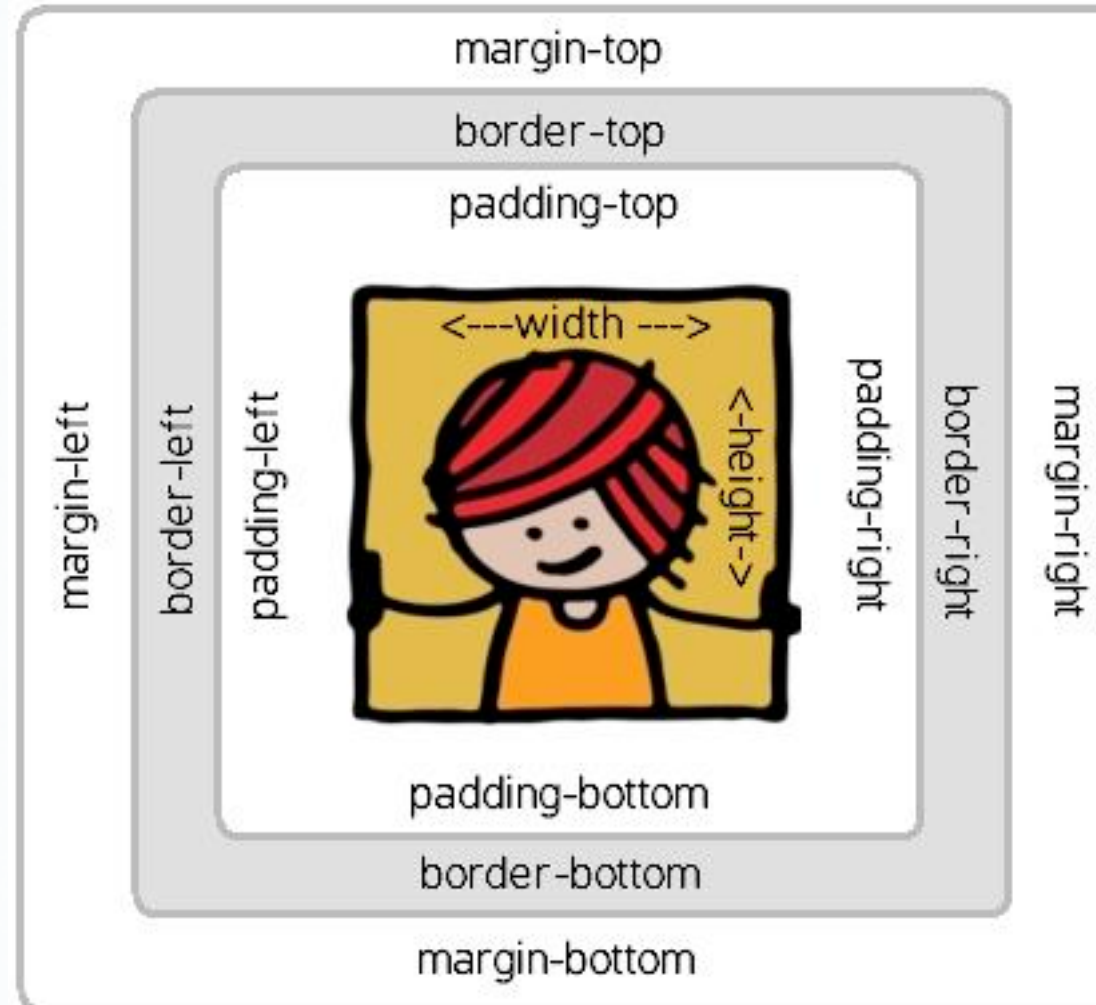
- CSS:

```
p { color: green; }  
p.grouped { color: blue; }  
p#unique { color: red; }
```

- результат:

First  
Second  
Third  
Just a regular p without id or class

# The box model





# margin

- Това са отстъпите, които поставяме от *външната* страна на елемента.
- Т.е. разстоянието на което ще отместим настоящия елемент, от всички обграждащи го
- Може да се задава поотделно за всяка страна на елемента (горе, долу, ляво, дясно)
- или за всички измерения наведнъж (композитно property)

# Пример

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: blue;
      }
      h1 {
        background-color: yellow;
        margin-top: 10px;
        margin-right: 20px;
        margin-bottom: 30px;
        margin-left: 40px;
        /* equivalent to: margin: 10px 20px 30px 40px; */
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```



# padding

- Това са отстъпите, които поставяме от *вътрешната* страна на елемента.
- Т.е. разстоянието на което ще отместим всички вложени елементи
- Може да се задава поотделно за всяка страна на елемента (горе, долу, ляво, дясно)
- или за всички измерения наведнъж (композиционно property)

# Пример

```
h1 {  
  padding-top: 10px  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 40px;  
}
```

```
h1 {  
  padding: 10px 20px 30px 40px;  
}
```



# border

- Използва се за рисуване на рамка на елемента
- Тъй като има дебелина, също влияе на отстоянието от всички други елементи
- Може да се задава поотделно за всяка страна на елемента (горе, долу, ляво, дясно)
- пак е композитно property, но има повече компоненти (дебелина, цвят, стил)

# Пример

```
h1 {  
  border-top-width: 20px  
  border-right-width: 20px;  
  border-bottom-width: 20px;  
  border-left-width: 20px;  
  border-style: solid;  
  border-color: red;  
}
```

```
h1 {  
  border: 20px red solid;  
}
```



# Think out of the box!

- Идеята за кутията идва от там че всички *html* елементи се представят като правоъгълници
- За съжаление обаче тази концепция не е съществувала още в началото, когато са били създадени първите *html* стандарти
- По онова време, фокусът е падал върху съдържанието (текстът) и не се е мислело много за обвивката на съдържанието (кутията)
- Съответно когато се оразмеряват елементите (с *width* и *height*) всъщност се оразмерява само съдържанието, а не кутията
- Резултатът - бързва се layout-а при ползването на отстъпи отвътре (padding)



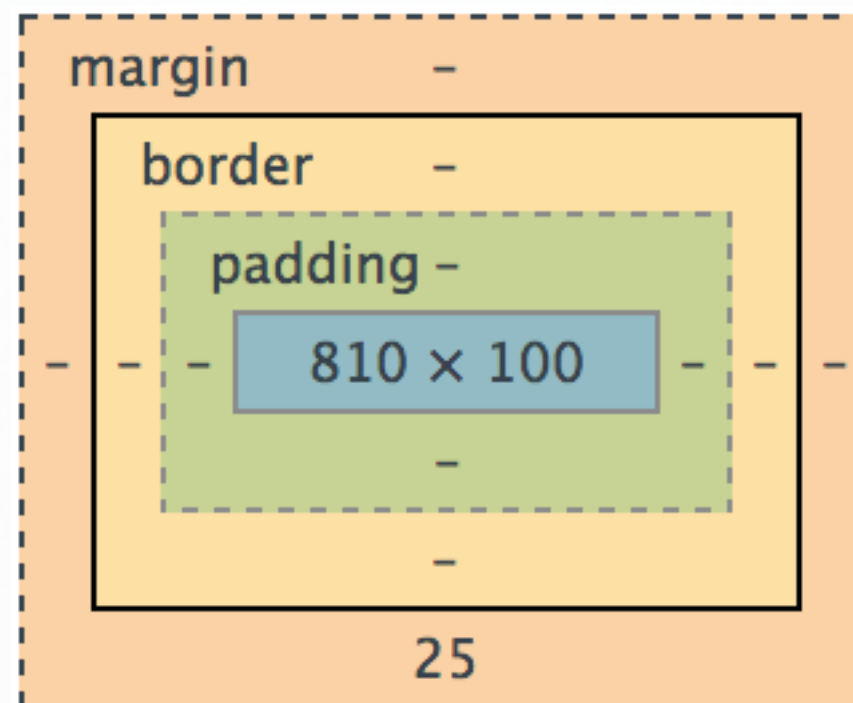
**SO WHAT YOU SAY IS THAT  
IF I SET BOTH WIDTH AND PADDING  
ON A DIV ..**

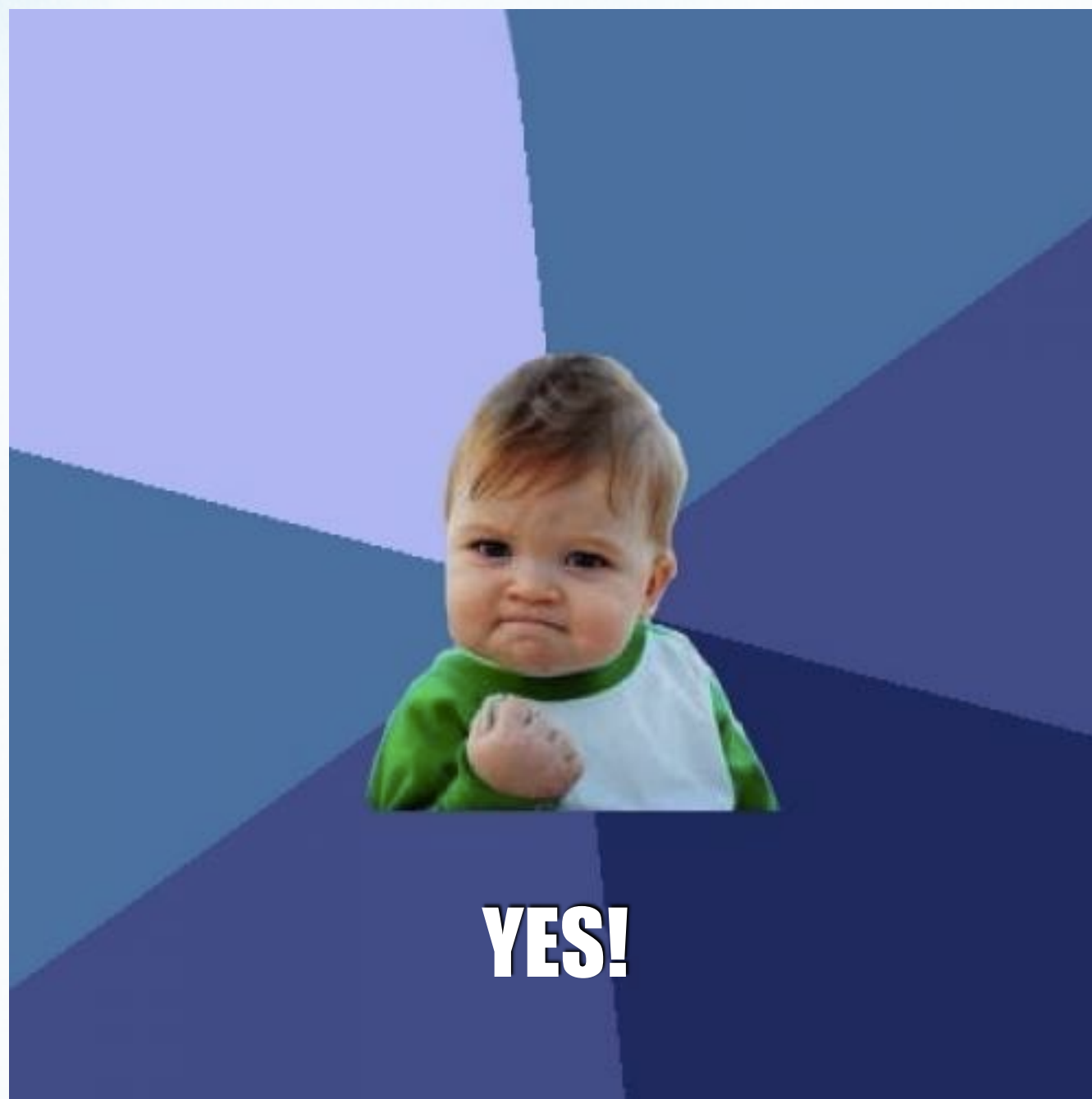
**IT'S SIZE WILL BE UNEXPECTED?**



Решението идва с CSS3:

```
* { box-sizing: border-box; }
```







# block vs inline

- Както вече казахме, всички *html* елементи се представят като правоъгълници
- Тези правоъгълници се разполагат по 2 основни начина на екрана: или се подреждат един след друг по хоризонтала, или един след друг по вертикала
- Правоъгълниците, които имат блоков характер (*блокови елементи*) се подреждат един под друг (заемат 100% от ширината на екрана)  
Пример: **div, p, h1, h2, .., h6, ul** и тн.
- Правоъгълниците с *inline* характер (*inline елементи*) се подреждат един до друг, стига да има място. Те заемат толкова пространство, колкото е тяхното съдържание  
Пример: **img, span, a**

# CSS display на HTML елементите

- Благодарение на CSS вече дори *native* поведението на *html* елементите може да бъде променено. Като например поведението им като *блокови* или *inline* елементи
- Това става благодарение на пропъртите *display*
- <https://css-tricks.com/almanac/properties/d/display/> (да се чете заедно със следващия линк по отношение на *display-table*)
- <http://colintoh.com/blog/display-table-anti-hero>
- <http://j4n.co/blog/Creating-your-own-css-grid-system>



# Домашно

- В хостинг папката ви си създайте папка **homeworks**, а в нея още по една папка за всяко от домашните до сега: L1, L2, L3 и т.н.
- Преместете файловете от съответното домашно в съответната папка
- Продължете работата си по електронната визитка и когато прецените че сте готови - качете кода в L3 папката
- Създайте папка L4 и в нея качете някой темплейт по ваш избор от този сайт:

<http://www.free-css.com/free-css-templates>

- изпратете ми линкове към страниците с домашни 3 и 4