



НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

Относно курса

Курса има общи условия, които можете да намерите тук:

courses.zenifytheweb.com

- Информация относно оценяване, изисквания, писане и предаване на домашни и т.н. - пак там

Контролни и изпит

- Тест на всеки 3-4 лекции
- Почивни дни: 2-те седмици в края на декември
- Финален изпит: ще се състои от въпроси, които се дават на интервю за работа, както и всичко, което сме взимали
- Проект: ще има и проект, който да направите през почивнивните дни (срока за предаване ще е 3-ти януари). Проекта и финалният изпит ще сформират заедно оценката от финалния изпит

Изисквания към кода, който предавате

- Спазване на конвенциите
HTML: http://www.w3schools.com/html/html5_syntax.asp
- Коректна идентификация
- Коректно именуване на файловете
- Още: <http://zenifytheweb.com/courses/#common-recommendations>

Срокове и комуникация

- Спазвайте сроковете, това също се отразява на оценката
- Ако някой има дори едно непредадено домашно, няма да получи оценка по “текущ контрол и домашни”
- Когато имате проблеми или “блокирате”, не се притеснявайте да пишете. Ако имам възможност - ще помогна.
- Присъединете се към Гугъл групата, за по-лесна комуникация:
 - група 1: <https://groups.google.com/forum/#!members/sa-fe-group1>
 - група 2: <https://groups.google.com/forum/#!members/sa-fe-group2>

Предаване

- Вместо да се предават лично, домашните трябва да се качат в GitHub (днес ще видим как)
- След като ги качите там, можете да проверявате дали са качени и на уебсайта на курса:
група 1: <http://zenifytheweb.com/courses/group1/index.html>
група 2: <http://zenifytheweb.com/courses/group2/index.html>
- повече за начина на предаване, можете да научите тук: <https://github.com/zzeni/swift-academy-homeworks>

Бонус задачи и точки

- Към някои от задачите за домашно ще има и допълнителни бонус задачи
- Условиата им ще бъдат качвани в гитхъб и всеки ще има възможност да се опита да ги направи
- Те няма да влияят върху оценката, но ще играят основна роля при избора на кандидати за отворените позиции по специалността
- Личната мотивация е много важна ;)

Домашните от миналия път

- Класически грешки:
 - грешен include на style.css
 - `<!DOCTYPE html>`
 - `& <=> &`
 - картинките
- за да можете да пишете на български:
 - `<meta charset="UTF-8">`
- errors vs warnings

Няколко CSS трик-а

- `ul { list-style-type: none; }`
- `h1 { text-transform: uppercase; }`
- `padding`
- `margin`
- `display: table;`
- `z-index: 999;`

GitHub

- Как се форк-ва GitHub repository
- Как се създава и управлява GitHub repository
- Ssh keys vs https
- GitHub Desktop

CSS3

За какво служи ?



Как се ползва?

За да включим CSS стилизиране в нашата HTML страница, използваме един от двата подхода:

- с линк (препоръчително):

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

- или просто inline:

```
<style>
```

...

```
</style>
```

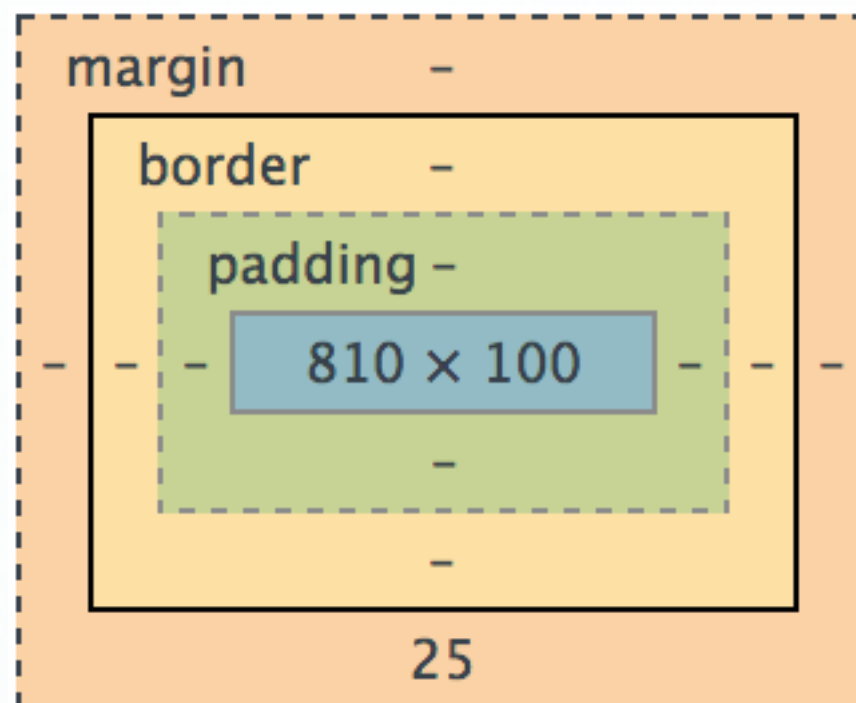
Синтаксис

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

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

...

padding, margin and border



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 (композиционно property)

Примери

```
<!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)

Примери

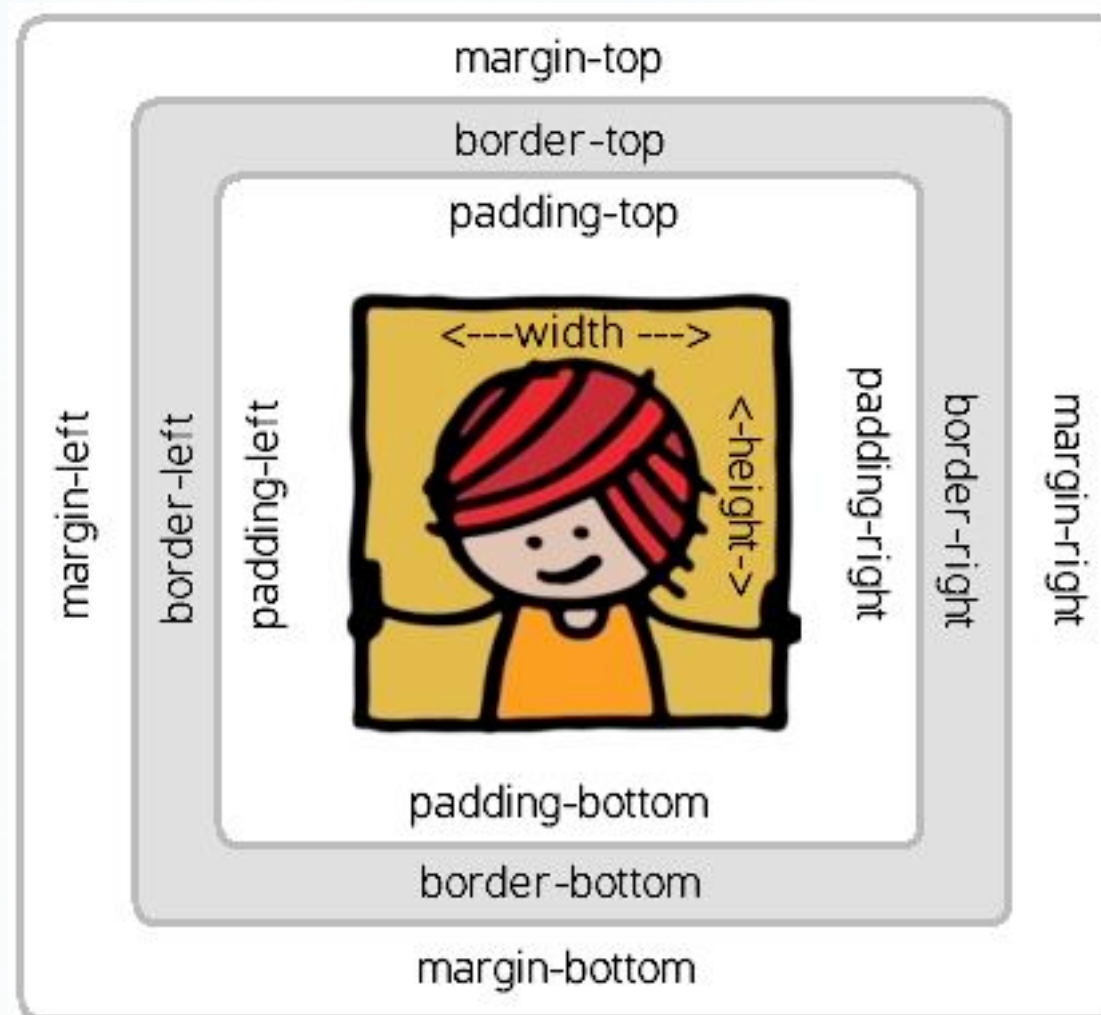
- `h1 {
padding-top: 10px
padding-right: 20px;
padding-bottom: 30px;
padding-left: 40px;
}`
- `h1 {
padding: 10px 20px 30px 40px;
}`

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)

Примери

- ```
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;  
}
```



It's easy, really !! :)

Какво е DOM

- Document Object Model
- От wikipedia:
The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML, and XML documents
- Ако си спомняте, споменахме че веб съдържанието, което браузърът получава, когато отваря дадена страница, се нарича документ
- т.е. html кода на нашата страница се интерпретира като документ, а неговото съдържание като DOM

Още за DOM

- DOM-а е съвкупността от всички елементи, тяхните атрибути и свойства
- Всеки един HTML елемент от нашата страница е част от DOM-а и всички елементи заедно сформират този DOM
- Самите браузъри парсват DOM-а до един сложен обект, който има множество полета и функции. Този обект се съхранява в глобалната променлива `document` и достъпът до него става чрез JavaScript, така че засега той няма да ни трябва
- DOM е конвенция, създадена от W3C! (<http://www.w3.org/DOM/>)

Какво е BOM

- Browser Object Model
- Аналогично на DOM, това е един сложен обект с множество полета и функции
- Докато DOM-а се определя еднозначно от самият уеб документ, BOM-а е различен за всеки браузър
- BOM-а съдържа цялата допълнителна информация за страницата, която не се съдържа в DOM-а, като например данните от request-а (headers от сървъра), настройки на клиента, бисквитки, информация за прозореца и т.н.
- Този обект се съхранява в глобалната променлива window и може да се достъпи чрез javascript
- https://en.wikipedia.org/wiki/Browser_Object_Model

Това, което трябва да запомним е че DOM-а е конвенция на W3C и че го ползваме за да намираме, достъпваме и манипулираме елементите в нашите html страници.

LET'S START USING THE DOM



Селектори

- Казахме, че DOM-а служи за идентифициране (намиране) на конкретен елемент от нашата html страница
- 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
- Използването на селектори е възможно, благодарение на DOM конвенцията
- http://www.w3schools.com/cssref/css_selectors.asp

id и **class** на елемента

- Вече знаем, че `id` е атрибут към `html` елемент, като го идентифицира по уникален начин
- сега видяхме как този атрибут, може да се използва за селектор
- Подобно на `id`, `class` е атрибут към `html` елемент, който обаче служи за групиране на сходни елементи и идентифицира уникално група от елементи
- запис в `html` таг-а:
``
- запис в `CSS`: `img.thumb { ... }`
или дори само: `.thumb { ... }`

Например

- HTML:
`<p id="uniqueue">First</p>`
`<p class="groupped">Second</p>`
`<p class="groupped">Third</p>`
`<p>Just a regular p without id or class</p>`
- CSS:
`p { color: green; }`
`p.groupped { color: blue; }`
`p#unique { color: red; }`
- резултат:

First
Second
Third
Just a regular p without id or class

Въпроси?

За следващия път

- НОВОТО задание, можете да намерите тук:
<https://github.com/zzeni/swift-academy-homeworks/tree/master/tasks/L3>

Recommended reading:

- <http://callmenick.com/categories/tutorials>