



НИ Е ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ



# Canvas





- canvas буквално означава платно за рисуване
- в HTML canvas е елемент, който се използва за динамично генерирано графично съдържание
- Или иначе казано - това е едно празно правоъгълно пространство, върху което да рисуваме нещо с javascript
- Документация и tutorials:
  - [http://www.w3schools.com/tags/ref\\_canvas.asp](http://www.w3schools.com/tags/ref_canvas.asp)
  - [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial)
  - <http://www.html5canvastutorials.com/tutorials/html5-canvas-element/>

# Параметри и особености

- Това е HTML5 елемент, който се поддържа от всички нови браузъри
- Ако браузърът не поддържа canvas, ще се изпише текста между двата тага (иначе текста не се вижда)
- Много важни параметри за canvas са ширина и дължина. За да зададем избрани от нас размери, можем да използваме HTML атрибутите width и height:

```
<canvas width="650px" height="200px">
```

```
  Your browser doesn't support canvas.
```

```
</canvas>
```

- добре е да се стараем canvas-а ни да е responsive !



# Интересни примери

- Часовник  
[http://www.w3schools.com/canvas/canvas\\_clock.asp](http://www.w3schools.com/canvas/canvas_clock.asp)
- Анимация с орбити и въртене  
<http://ocanvas.org/demos/4>
- 3D анимация с кубове  
<http://cssdeck.com/labs/html5-canvas-3d-cubes>





# Embedded content

# HTML5 Video & Audio

- video и audio са html5 мултимедиа елементи, които дават възможност за embed-ване (вмъкване) и стрийм-ване (изпълнение) на видео и аудио съдържание
- Важно: поддържани файлови формати: [https://developer.mozilla.org/en-US/docs/Web/HTML/Supported\\_media\\_formats](https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats)
- Важно: дори и елемента да се поддържа от браузъра, не всички формати ще работят. Затова е добре да имаме по няколко формата:

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
  Your browser does not support the video tag.  
</video>
```



- source - път към медия файловете (може да са повече от един)
- track - допълнително съдържание, например субтитри
- атрибути: controls, autoplay, poster
- Още референции:
  - [https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using\\_HTML5\\_audio\\_and\\_video](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML5_audio_and_video)
  - [http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp)
  - [http://www.w3schools.com/html/html5\\_audio.asp](http://www.w3schools.com/html/html5_audio.asp)

# Plug-ins

- Това са елементи, които използват външна програма (плъгин за браузера), за да може да се render-не някакво съдържание. Такива елементи са **embed** и **object**
- Например, преди да се появят аудио и видео таговете, за ембедване на мултимедийно съдържание и флаш се използваха тези елементи, които пък зареждаха плъгини като **flash media player**
- Разликата между **embed** и **object** е най-вече в това, че **object** е документиран още в html4, а **embed** - едва в html5. На практика обаче и двата елемента се поддържат от всички стари браузъри
- мое лично мнение: ползваме **audio** и **video**, където може (когато форматът на файла се поддържа) и **embed** - когато трябва да се използва 'външна' програма (плъгин) за този файлов формат



# iframe

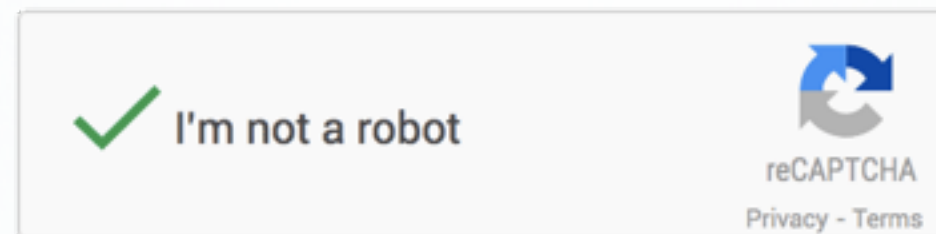
- Това е интерактивна рамка, съдържаща вмъкнато HTML съдържание
- Използваме го за embed на html съдържание
- Чрез iframe можем да вмъкнем съдържание от други уебсайтове в нашият
- Реално се получава изтегляне и рендериране на другия html документ вътре в нашия html документ (страница в страницата)
- [http://www.w3schools.com/html/html\\_iframe.asp](http://www.w3schools.com/html/html_iframe.asp)

W3C спецификация:

<http://www.w3.org/TR/html5/embedded-content-0.html>



# CAPTCHAS



- Completely Automated Public Turing test to tell Computers and Humans Apart
- Т.е. това е тест, който да попречи на ботове (програми, които се представят за хора) да събмитват информация в нашата страница
- Най-често използваме публичен captcha API за картинките
- при попълване на captcha полето от потребителя и събмитване на формата, captcha параметърът се изпраща заедно с останалите параметри към бекенд-а
- Бекенда има за отговорност да изпрати към captcha provider-а (API-а) въведената стойност, за да я верифицира



# Въпроси?

# Примери

<http://zenifytheweb.com/courses/lessons/lesson21/example/index.html>