

# JS, DOM, CSS

Davide Morelli

CI 2018

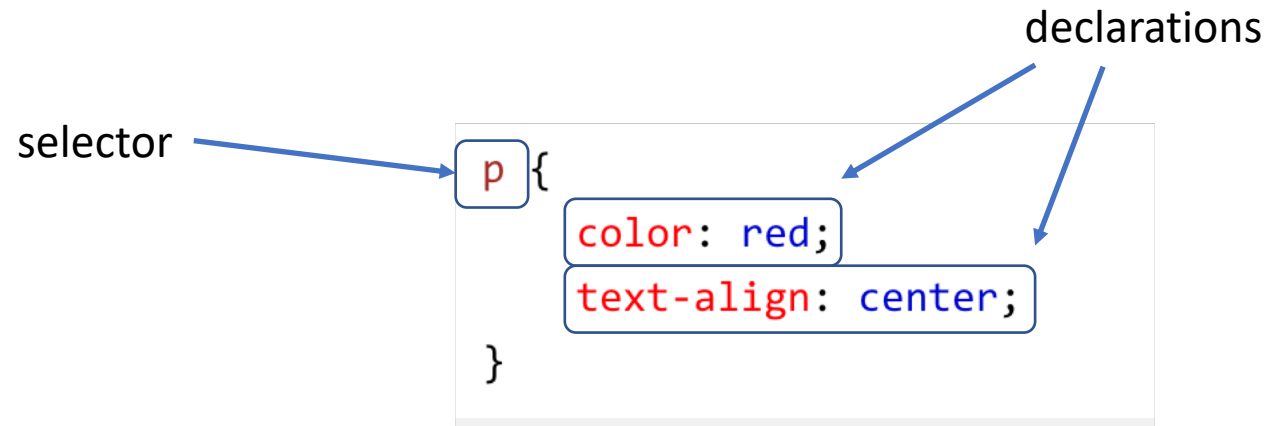
# CSS

- Syntax

```
p {  
    color: red;  
    text-align: center;  
}
```

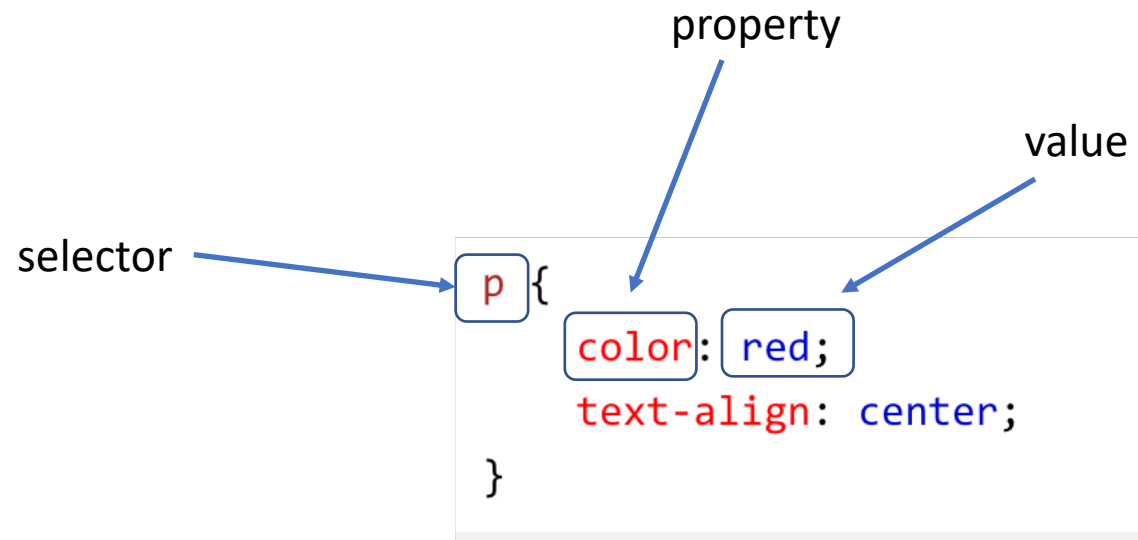
# CSS

- Syntax




# CSS

- Syntax



# 3 ways to add style sheets

- External style sheet



```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

- Internal style sheet



```
<head>  
<style>  
  body {  
    background-color: linen;  
  }  
  
  h1 {  
    color: maroon;  
    margin-left: 40px;  
  }  
</style>  
</head>
```

- inline



```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

# CSS selectors

- Select by tag name:
  - E.g.: `p {color: black}`
- Select by ID: `#`
  - E.g.: `#paragraph1 {color: red}`
- Select by class: `.`
  - E.g.: `.centeredText {text.align: center}`

# CSS selectors

- Select by attributes:
  - [attr]: presence of attribute *attr*
  - [attr=value]: the value of attribute *attr* is *value*
  - [attr^=value]: the attribute *attr* starts with *value*
  - [attr\$=value]: the attribute *attr* ends with *value*
  - [attr\*=value]: the attribute *attr* contains the string *value*
  - [attr~=value]: the attribute *attr* contains the string *value* as one of its space separated elements

# CSS selectors

- Pseudoclasses
  - *Elements selected only when in a certain state*
  - *:active, :hover, :visited, etc.*



# Combinations of selectors

- A, B: elements matching A and/or B
- A B: elements that match B that are descendant of A
- A > B: elements that match B that are direct child of A
- A + B: elements that match B that is the next sibling of A
- A ~ B: elements that match B that is one of the next siblings of A

# Cascading order

## 1. Importance

1. !important
2. External and internal style sheets
3. Browser default

## 2. Specificity

1. Inline
2. ID
3. Class, attribute, pseudo-class
4. Element, pseudo-element

## 3. Source order

# Cascading order

Selector	Thousands	Hundreds	Tens	Ones	Total specificity
h1	0	0	0	1	0001
h1 + p::first-letter	0	0	0	3	0003
li > a[href*="en-US"] > .inline-warning	0	0	2	2	0022
#identifier	0	1	0	0	0100
No selector, with a rule inside an element's <a href="#">style</a> attribute	1	0	0	0	1000

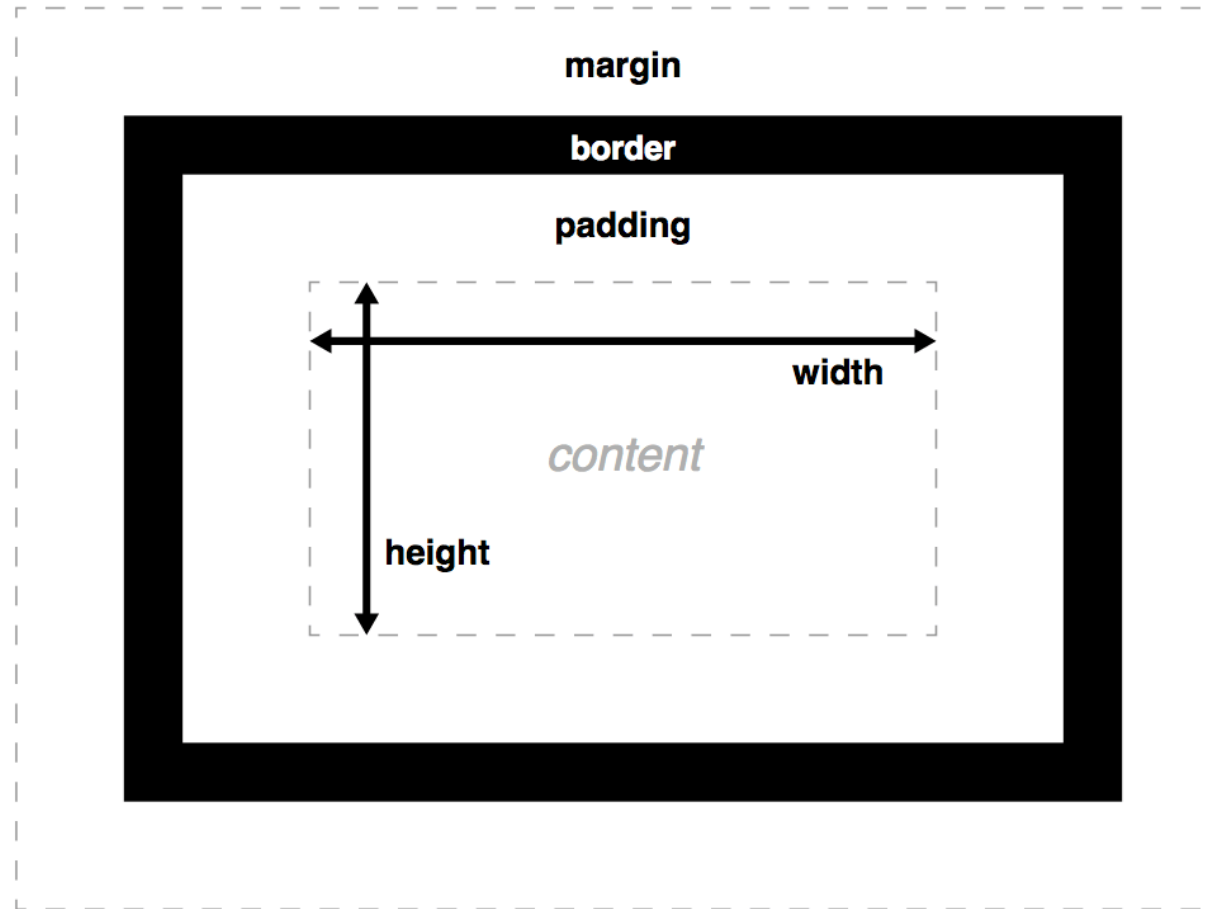
# Inheritance

- Some styles have inheritance, some don't
- E.g.
  - font-family, color: yes
  - margin, padding, border, background-image: no
  - Common sense (a.k.a. check reference)
  - <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

# Control inheritance

- inherit: sets value of property to its parent
- initial: set to browser default
- unset: back to initial or inherit
- revert: rolls back removing styles from current origin

# The box model



# The box model

- overflow
  - auto
  - hidden
  - visible
- see example

# The box model

- boxes types:
  - block
  - inline
  - inline-block
- see example



# Colors

- Names

```
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

- RGB

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
```

- #

```
<h1 style="background-color:#ff6347;">...</h1>
```

- RGBA

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
```

- HSL

```
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```

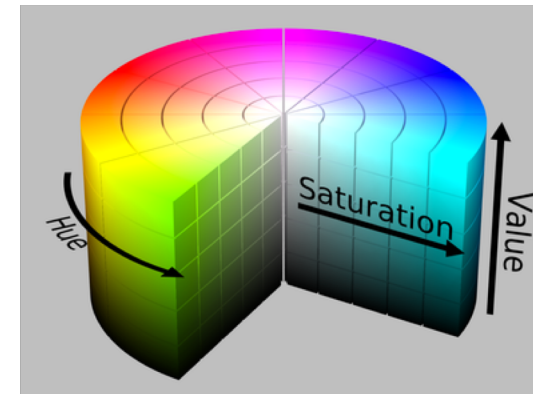
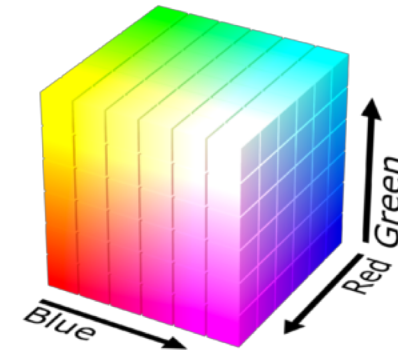
- HSLA

```
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

See [https://developer.mozilla.org/en-US/docs/Web/CSS/color\\_value](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value)

# Color models

- RGB
  - Red Green Blue
- HSL (cylindrical model)
  - Hue
    - circular, starts at red (0°), goes through green (120°) then blue (240°)
  - Saturation
    - distance from axis, that has white, grey, black
  - Value
    - lightness



# JS+DOM+CSS

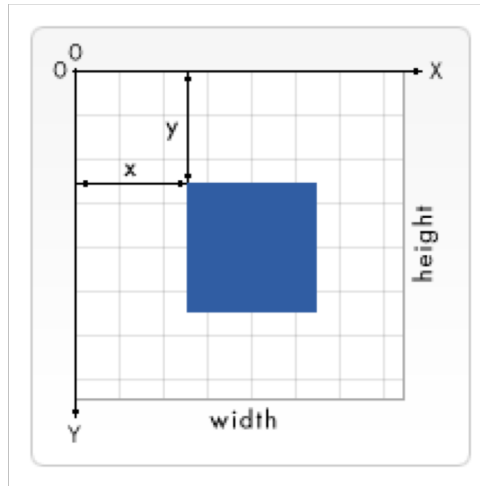
- bouncing divs example

# Canvas

- HTML5 element
- provides a 2D graphics context, or access to WebGL (close to OpenGL ES 2.0)
  - no paint, retention model
- see
  - [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API)
  - <http://curran.github.io/HTML5Examples/>

# canvas

- can (should) have fallback content
- must close with `</canvas>`
- coordinate space



# canvas

- see examples:
  - paths, Path2D, colors, strokeStyle, line styles, line caps, line joins, shadow, text, images, save() and restore(), translate(), rotate(), scale(), clipping
    - transformation matrix:
      - a: horizontal scaling
      - b: horizontal skewing
      - c: vertical skewing
      - d: vertical scaling
      - e: horizontal moving
      - f: vertical moving
- gallery example
- animation example


$$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

# CSS animations

add animation properties to the declarations. It must contain duration and name. Can have multiple animations simultaneously.

keyframes define what properties will change and how

```
1  p {  
2      animation-duration: 3s;  
3      animation-name: slidein;  
4  }  
5  
6  @keyframes slidein {  
7      from {  
8          margin-left: 100%;  
9          width: 300%;  
10     }  
11  
12     to {  
13         margin-left: 0%;  
14         width: 100%;  
15     }  
16 }
```



# animations event listeners

```
1  var element = document.getElementById("watchme");  
2  element.addEventListener("animationstart", listener, false);  
3  element.addEventListener("animationend", listener, false);  
4  element.addEventListener("animationiteration", listener, false);  
5  
6  element.className = "slidein";
```