

# CSS

Lab. Bases de Dados e Aplicações Web  
MIEIC, FEUP 2016/17

Sérgio Nunes

# Summary

- Quick Overview
- The CSS Language
- CSS Selectors & Properties
- The Box Model
- CSS Positioning
- Overview of CSS3

# Quick Overview

# The Big Picture

- Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents.
- It is a client-side technology used to format the presentation of web documents.
- Can exist within an HTML document or be referenced by it.

[Sign Up](#) [Facebook helps you connect and share with the people in your life.](#)

Email  Password [Login](#)  
 Keep me logged in [Forgot your password?](#)

[Wall](#) [Info](#) [Photos \(64\)](#) [Resources](#) [Stories](#) [Press](#) [Video](#) [Facebook Live](#) [More](#)

**35,370,906**  
people like this

[Likes](#) [See All](#)

 Facebook Australia & New Zealand  
 U.S. Politics on Facebook  
 Democracy UK on Facebook  
 Facebook for Android  
 Facebook Safety

[Create a Page](#) [Share](#)

**Facebook** [Like](#)  
(Product/Service)

**Wall**

 **Facebook**  
Take a look at how news of the Japan earthquake and tsunami spread on Facebook. For more resources and ways to help, check out the [Global Disaster Relief](#) on Facebook page.

 **Japan Earthquake & Tsunami**  
Scroll through this image gallery to see how news of the earthquake and tsunami in Japan spread on Facebook via status messages. Now let's spread information about how to help the relief effort.  
By [Global Disaster Relief on Facebook](#)  
Photos: 10

 9 hours ago · Like · Comment  
 Hassan Issa, Dorothy Stuart, Vishnu Kunwade and 15,711 others like this.  
 View all 2,648 comments

 **Facebook**  
Our thoughts are with all of those impacted by the devastating earthquake and tsunami in Japan. Some people, like Tiffany Brunson, are using Facebook to get updates on loved ones. Her sister in Tokyo let Tiffany and her family in North Carolina know she was okay through status updates. Learn how you can help, too, on the [Global Disaster Relief](#) on Facebook Page.

  Friday at 12:51pm · Like · Comment  
 Louis Edmiston, R/G/B, TommyShadwellBlast, Sérólfalibundfrómhéðiðháðen and 38,544 others like this.  
 View all 5,037 comments

 **Facebook**  
We're taking more steps to help you stay safe online. Learn more in an announcement on the [Facebook Safety](#) page at <http://on.fb.me/12NHVu>, and join a discussion on the topic on [Facebook Live](#) from the White House at 12:20 pm EST today at <http://on.fb.me/1fB8nJ>.  
March 10 at 7:51am · Like · Comment  
 Circle Cayugan, Valen Ley, Marina Sofianou and 11,351 others like this.  
 View all 5,403 comments

 **Facebook**  
Story of the Week: A 4-year-old boy is diagnosed with leukemia after a picture his father posted of him on Facebook caught the attention of a concerned friend.

 **Boy, 4, Diagnosed With Leukemia After Picture Is Posted on Facebook**  
[www.foxnews.com](http://www.foxnews.com)  
A British doctor realized that a 4-year-old boy had leukemia after seeing his picture on Facebook — and now the boy's father is crediting the picture with saving his son.

 March 9 at 12:44pm · Like · Comment  
 Julian FrââmCisqâdâ Pidilîmâ SâlîzZâr, Muhammad Saman Al-Bahrâz, Young Money Message and 15,818 others like this.  
 View all 3,576 comments

Facebook

Email  Password  Forgot your password?

Keep me logged in [Forgot your password?](#)

Sign Up Facebook helps you connect and share with the people in your life.



Wall (2)  
Jobs (3)  
Photos (26)  
Reactions (3)  
Events (5)  
Places (2)  
Videos (2)  
Facebook Lite (2)  
Notes (1)  
A Page About (2)  
Mobile (2)  
Groups (2)  
Like New York (2)  
Links (2)

Most  
Less

11,370,006 people like this.  
[See All](#)

Likes

 Facebook Safety

 Know Issues on Facebook

 Facebook Australia & New Zealand

 Facebook for Android

 Downing UK on Facebook

Create a Page  
 Share

**Facebook**

[Like](#)  
Product/Service

•

•

•

•

Wall



Facebook Take a look at how some of the Japan earthquake and tsunami spread on Facebook. For more information and ways to help, check out the [Global Disaster Relief on Facebook](#) page.

**Japan Earthquake & Tsunami**  
Scroll through this image gallery to see how news of the earthquake and tsunami in Japan spread on Facebook via status messages. Now let's spread information about how to help the relief effort.  
By [Global Disaster Relief on Facebook](#)  
Photos (10)  
[Share](#) [Like](#) [Comment](#) [View Feedback \(18,371\)](#) [Hide Feedback \(18,371\)](#)  
+ [Anand Thakur](#), [All Done](#), [Edward Hall](#) and [11,704 others](#) like this.  
+ [View all 1,046 comments](#)



Facebook The thoughts are with all of those impacted by the devastating earthquake and tsunami in Japan. These people, the White House, are using Facebook to raise awareness and donations. We urge you to help for Tiffany and her family in North Carolina before the snow melts through status updates. Learn how you can help here on the [Global Disaster Relief on Facebook](#) page.

**Tiffany at 12:25pm** [Like](#) [Comment](#) [View Feedback \(13,582\)](#) [Hide Feedback \(13,582\)](#)  
+ [TiffanyMiddendorf](#), [MicheleLynnFriedman](#), [Athena](#), [Patty](#), [Jill](#), [Diane](#) and [18,740 others](#) like this.  
+ [View all 1,026 comments](#)



Facebook Everyone deserves a safe place to live, work and play. Learn more in an announcement on the [Facebook Jobs](#) page or [FacebookJobs2011a](#), and join a discussion on the topic on [Facebook Live](#) from the White House at 12:30 pm EST today at [Facebook.com/WhiteHouse](#).

**March 11 at 7:35am** [Like](#) [Comment](#) [View Feedback \(14,756\)](#) [Hide Feedback \(14,756\)](#)  
+ [Crista Conner](#), [Vivian Lee](#), [ShaneSoftware](#) and [11,351 others](#) like this.  
+ [View all 1,542 comments](#)

## HTML + CSS

HTML only

# Original Vision

- HTML is for content and structure.
- Presentation is left to the user allowing local configurations. Each user has its settings.
- However, web authors wanted to control presentation. Browsers introduced tags and attributes.
- Over time, the HTML language "degenerated" to also include presentation markup.

`<b>...</b>`

`<i>...</i>`

`<blink>...</blink>`

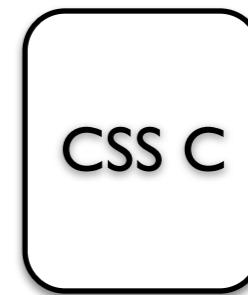
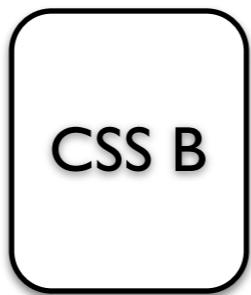
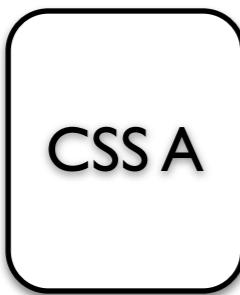
`<font size="x">...</font>`

# Problems

- Bloated code. No "separation of concerns".
- Difficult to redesign.
- Difficult to support different devices.
- Content formatted for a given medium/form.
- Hard coded decoration and layout.

# Motivation for CSS

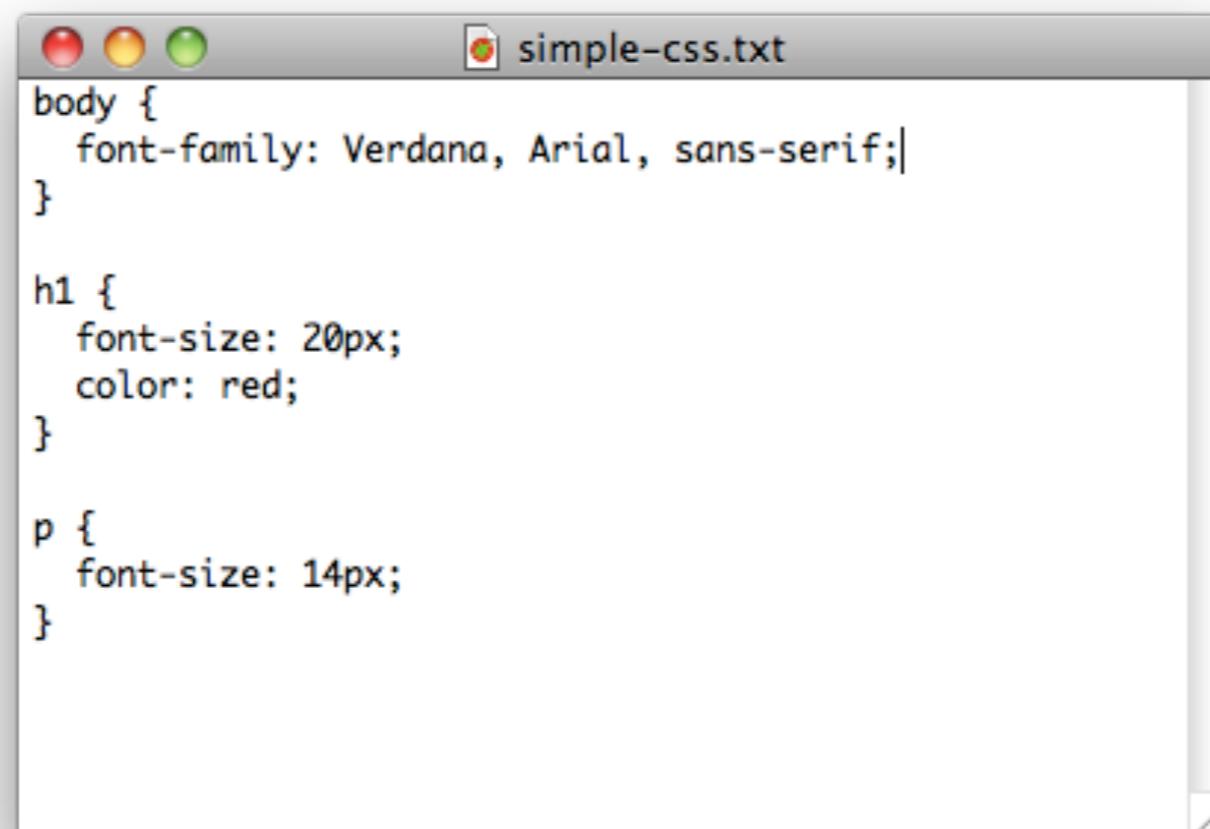
- Separate structure from presentation.
- Different tools for different skills.  
Content versus Design.
- Avoid redundancy. Centralized rules for presentation.  
Control multiple HTML documents with one CSS file.
- Easier support for different devices (e.g. mobile, desktop, print, web).
- Simpler maintenance.



From "css Zen Garden: The Beauty in CSS Design"  
<http://www.csszengarden.com/>

# CSS Document

A CSS document is a text document containing CSS rules. CSS rules can also be directly included in an HTML document.



```
simple-css.txt
body {
    font-family: Verdana, Arial, sans-serif;
}

h1 {
    font-size: 20px;
    color: red;
}

p {
    font-size: 14px;
}
```

# CSS Specifications

- 1994 - Cascading Style Sheets at CERN.  
<http://www.w3.org/People/howcome/p/cascade.html>
- 1996 - CSS 1 published as a W3C Recommendation.
- 1998 - CSS 2 published as a W3C Recommendation.
- 2000 - CSS 3 published as W3C Working Draft.
- 2002 - CSS 2.1 published as W3C Working Draft.

# The CSS Language

# Applying CSS

# Cascading Style Sheets

- Three different levels of styles (cascading):
  - Inline - element level.
  - Internal - document level.
  - External - site level.

# Inline Styles

- HTML elements can be directly styled using the `style` attribute.

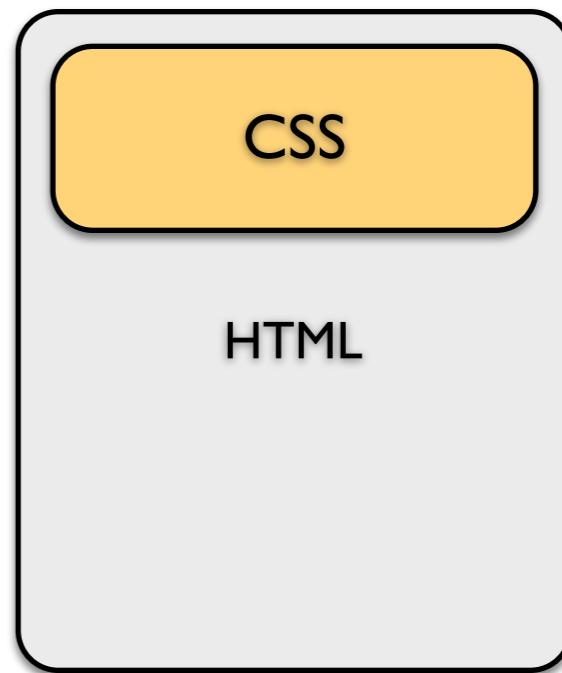
```
<h1 style="color: red;">A Title</h1>
```

```
<em style="font-weight: bold;">this is important</em>
```

# Internal Styles

- Style sheets can be embedded into the head element of a HTML document.

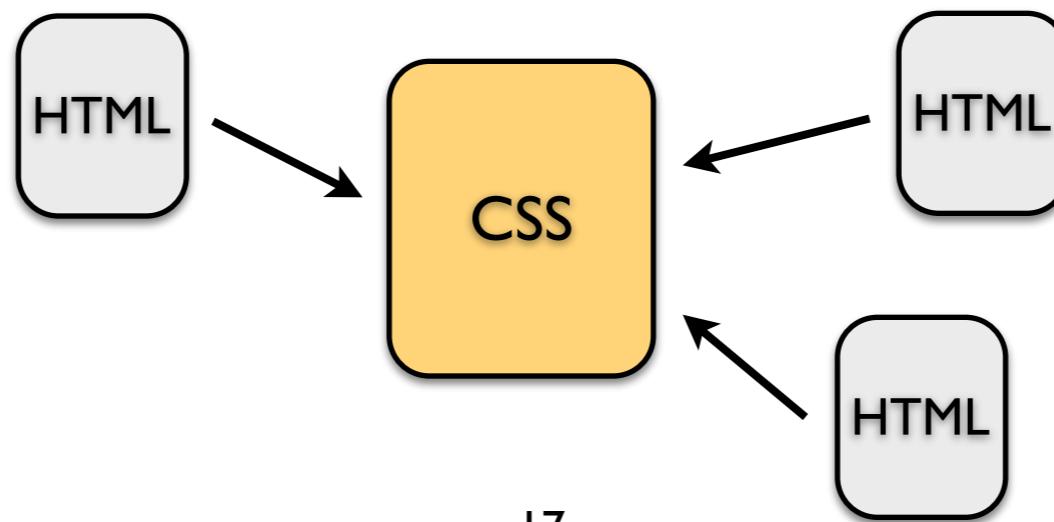
```
<head>
  <style type="text/css">
    h1 {
      color: red;
    }
    em {
      font-weight: bold;
    }
  </style>
</head>
```



# External Styles

- Both inline and internal rules cannot be reused across a site.
- Using external styles, CSS rules are included in a specific CSS file and linked to HTML documents.

```
<link rel="stylesheet" type="text/css" rel="path/to/file.css">
```



# Priorities

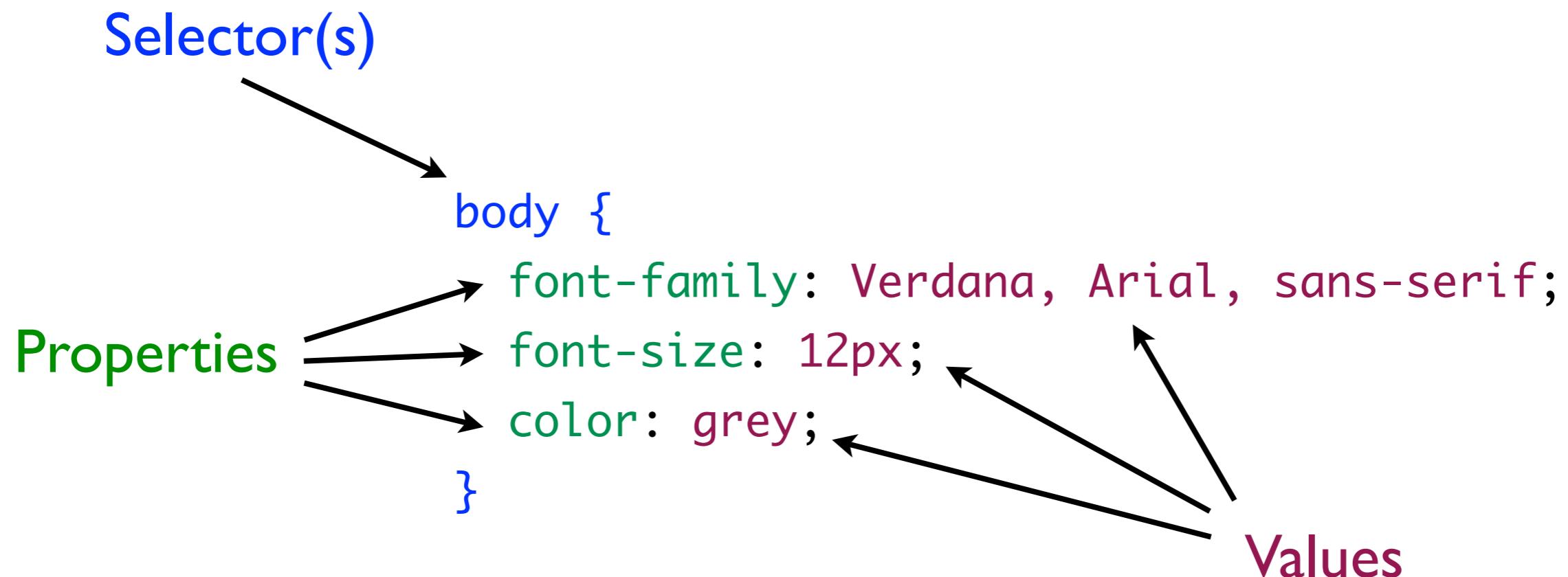
- When applying styles to documents, the following priority is followed:
  - Browser defaults.
  - External style sheets.
  - Internal styles.
  - Inline styles.

# CSS Syntax

# CSS Files

- A CSS file (style sheet) consists of a list of rules (or statements).
- Each rule consists of one or more selectors and a declaration block, enclosed in curly brackets.
- Each declaration consists of a property, a colon (:), a value and a semicolon (;).

# Anatomy of a Rule



# Coding Conventions

- CSS files can have comments that are ignored by the browser. Useful for documenting or organizing code.

```
/* This is a comment */          /* This is a  
multiline comment */
```

- Layout CSS rules in multiple lines for improved readability. Use a single line in special cases.

```
h1 {  
    font-size: 18px;  
    text-align: center;  
}  
  
.alert { color: red; }
```

# CSS Selectors

# CSS Selectors

A CSS selector identifies an object on a web page to which CSS declarations are applied.

```
body {                                div.nav p strong {  
    color: red;                      color: yellow  
    font-family: Arial;             }  
    font-size: 18px;  
}  
                                
```

# Type Selectors

- HTML elements are used as selectors.
- Should be used when you want to affect every instance of a HTML element.

```
h1 { color: red; }  
p { color: black; }  
img { border: none; }  
div { border: 1px solid black; }
```

# Class Selectors

- How to change a specific instance of an element?  
For instance, how to style the 2nd paragraph only?
- It is possible to define classes in CSS and apply them to HTML elements. A given class can be used multiple times on the same document.

```
.current {}  
.title {}
```

CSS file

```
<h2 class="title">...</h2>  
<p>...</p>  
<p class="current">...</p>  
<p class="current">...</p>
```

HTML document

# Using Class Selectors

- Class selectors can either be applied to all elements of a given class, or only applied to a specific HTML element of that class.

```
.current {}  
.title {}
```

```
p.current {}  
h2.current {}  
p.title {}
```

# Multiple Classes

- A single HTML element may be associated with multiple classes, separated by spaces.

```
.current {}
```

```
.note {}
```

CSS file

```
<p class="current">...</p>
<p class="current">...</p>
<p class="current note">...</p>
```

HTML document

# ID Selectors

- ID selectors are similar to class selectors, except that they can only be applied to exactly one element on a page. One ID per document.
- ID selectors are defined in CSS with '#', and applied to elements with the 'id' attribute.

```
#active {}
```

```
p#footer {}
```

```
div#footer {}
```

CSS file

```
<p id="active">...</p>
```

```
<div id="footer">...</div>
```

HTML document

# ID and Class Naming

- ID and Class names can contain lowercase letters, uppercase letters, digits, underscores and dashes.  
Cannot start with digits.

.current #Header1

.itemTitle #item-title

- Names should define function, not appearance.
  - Good: header, footer, active, note.
  - Bad: italic, red, big.

# Descendent Selectors

- Can be used to select an element based on its position within the document's HTML structure, e.g. paragraph inside div 'footer'.

a {}

p a {}

p.current a {}

ul li a {}

#footer .note em a {}

# Link Pseudo Class Selectors

- Link pseudo class selectors are used to select links in a number of different states. Links have 4 states:
  - normal — standard, unvisited link.
  - visited — visited link.
  - hover — when the cursor is over the link.
  - active — while the link is being clicked.

`a:link {}`  
`a:visited {}`  
`a:hover {}`  
`a:active {}`

`a.note:link {}`

May be combined  
with class selectors.

# Pseudo Element Selectors

- Two types of pseudo element selectors

<code>:first-line</code>	<code>:before</code>
<code>:first-letter</code>	<code>:after</code>

- `first-line` — selects first line of element.
- `first-letter` — selects first letter of element.
  
- `before` — inserts content before the element selected.
- `after` — inserts content after the element selected.

```
p:first-line { text-transform: uppercase; }
```

Hello world, this is my first CSS rule using pseudo element selectors.

HELLOWORLD, THIS IS my first CSS rule using pseudo element selectors.

```
p:first-letter { font-size: 2em; }
```

Hello world, this is my second CSS rule using pseudo element selectors.

HHello world, this is my second CSS rule using pseudo element selectors.

```
h1:before { content:"Chapter: "; }
```

Introduction

Chapter: Introduction

```
a:after { content:attr(href); }
```

U.Porto

U.Porto <http://www.up.pt>

# Grouping Selectors

- Different selectors may be grouped in a single CSS rule, using commas.

```
ul li { color: red; margin: 5px; }  
p { color: red; }  
h1 a { color: red; }  
.new { color: red; }
```

```
ul li, p, h1 a { color: red; }  
ul li { margin: 5px; }
```

# Universal Selector

- The universal selector matches the name of any element. It selects any single element from the document.

```
* { font-family: Verdana, sans-serif; }
```

```
* {  
  margin: 0;  
  padding: 0;  
}
```

# Child Selectors

- A child selector matches when an element is the direct descendant (child) of some element.

```
p > a {}  
ul > li {}
```

# First Child Selectors

- The first child selector matches an element that is the first child of some element. The first child of any parent, not the first child of an element.

```
p:first-child {}
```

```
p:first-child em {}
```

# Adjacent Selectors

- Adjacent (sibling) selectors matches an element that immediately follows another element at the same level.

`strong + em {}`

Matches em elements that follow strong elements

`ul + p {}`

Matches paragraphs that immediately follows lists

# Attribute Selectors

- Attribute selectors matches elements that have specific attributes defined.
- Attributes may match in four ways:
  - `[att]` — match when attribute `att` is set.
  - `[att=val]` — match when attribute `att` has value `val`.
  - `[att~=val]` — match when attribute `att` includes the `val` keyword among a space separated list.
  - `[att|=val]` — match when attribute `att` includes the `val` keyword among a dash separated list.

h1[title]

```
<h1 title="Portugal">...</h1>  
<h1 class="some">...</h1>
```

h1[title="Porto"]

```
<h1 title="Porto">...</h1>  
<h1 title="Portugal">...</h1>
```

h1[title~="Porto"]

```
<h1 title="Porto Portugal">...</h1>  
<h1 title="Porto-Portugal">...</h1>
```

h1[title|= "Porto"]

```
<h1 title="Porto Portugal">...</h1>  
<h1 title="Porto-Portugal">...</h1>
```

# Advices

- Don't "over DIV" your page.  
Keep it simple.
- Reuse HTML native elements if adequate.  
Look at the new structural elements in HTML5.
- Don't "over class" your CSS.  
Use advanced selectors.
- Group selectors in common rules.

# Text Formatting

# Color Property

- The color property sets the foreground color of an element (i.e. text).
- The color property can be set by different ways: keyword, RGB code, Hex code.

**Keyword:**

black  
yellow  
red  
blue

**RGB code:**

rgb(100%, 40%, 0%) = rgb(255, 102, 0)  
rgb(40%, 20%, 20%) = rgb(102, 51, 51)

**Hex code:**

#326432  
#0088ff  
#000000

# Text Properties

- text-indent: specifies the indentation of the first line of the text in a block.
- text-align: describes how inline content of a block container is aligned (left, right, center, justify).
- text-decoration: decorations added to the text (underline, overline, line-through).
- text-transform: controls the capitalization effects of text (capitalize, uppercase, lowercase).

# Text Properties

- letter-spacing: spacing behavior between text characters.
- word-spacing: spacing behavior between words.
- line-height: minimal height of lines within a block element.
- vertical-align: specifies the vertical position of content within a block element.

# Font Properties

- font-family: prioritized list of font family names and/or generic names.
- font-style: selects between normal and italic faces within a font family.
- font-size: specifies the font size.
- font-weight: select the weight of the font.
- font-variant: specifies the small-caps variation.

# Font Families

- Two types of font family names:
  - `<font-family>` — the name of a font family of choice (e.g. Helvetica).
  - `<generic-family>` — the name of a generic family name: serif, sans-serif, cursive, fantasy and monospace.

# Generic Font Families

- Serif fonts has finishing details (i.e. serifs) on the ends of symbols (e.g. Times, Georgia, Garamond).
- Sans-serif fonts don't have serifs at the end of strokes (e.g. Helvetica, Verdana, Trebuchet MS).
- Monospace fonts have equal width symbols (e.g. Courier).

Ai

Ai

Ai

# Web-safe Fonts

- Fonts that are present in a wide range of browsers and operating systems.
- Microsoft's "Core fonts for the Web" had an important impact in web typography.
- High penetration fonts: Arial, Courier New, Georgia, Times New Roman, Trebuchet, Verdana.

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Arial

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Georgia

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Trebuchet

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Courier

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Times New Roman

The Quick Brown  
Fox Jumps Over  
The Lazy Dog.

g

abcdefghijklmnopqrstuvwxyz0123456789 [](){}/\<>?

Verdana

# Alternatives

- How to use non web-safe fonts?
  - Image replacement: high control, low accessibility, higher bandwidth.
  - Flash-based: selectable text, requires Flash.
  - Others: JavaScript + SVG, CSS3's font-face.
- External services: Google Fonts, TypeKit.

# Font Sizes

- Font sizes can be specified with CSS using:
  - `<absolute-size>` - large / medium / small, ...
  - `<relative-size>` - larger / smaller.
  - `<length>` - points, ems, pixels, centimeters, ...
  - `<percentage>` - relative to parent's size.
- The font size property is inherited from parent.

# CSS Units

# CSS Units

Unit	Type	Example
px (pixels)	length	width: 744px;
em (ems)	length	margin-left: 1.25em;
% (percent)	length	left: 34%;
pt (points)	length	font-size: 12pt;
in (inches)	length	margin-top: .75in;
cm (centimeters)	length	margin-top: 1.905cm;
xx-small ... xx-large	font-size	font-size: large;
rgb(r,g,b)	color (decimal)	background-color: rgb(221,204,187);
#rrggbb	color (hexadecimal)	background-color: #ddccbb;
#rgb	color (hexadecimal)	background-color: #dcb;

# Screen Display Units

- **px** (pixels) — Pixels are absolute units, equal to one pixel on the user's screen display; always expresses as an integer.
- **em** (ems) — An em is equivalent to the greatest possible height of a letter in the applicable font and size combination. Usually expressed as a floating-point value.
- **%** (percent) — Percentage units are computed relative to some baseline measurement, which varies according to property and context. Floating-points are allowed.

# Print-Friendly Units

- **pt** (points) — Type is traditionally measured in points. A point is roughly equivalent to 0.0352 cm, yielding slightly less than 28.35 points per centimeter.
- **in** (inches) — One inch is defined as being equal to 2.54 cm.
- **cm** — Centimeters.

# Color Units

- CSS supports a three-channel (red, blue, green) color space with 8 bits of depth per channel, i.e.  $2^{24}$  possible colors.
- Three styles of notation are available.
  - `rgb(r,g,b)` — Three channels, decimal. Each channel takes a range of 0-255.
  - `#rrggbb` — Three channels, hexadecimal. Each channel takes a range of 00-ff.
  - `#rgb` — Three channels, hexadecimal, reduced depth. Each channel takes a range of 0-f. `#6fc` = `#66ffcc`

# CSS and Graphics

# CSS and Graphics

- It is possible to format graphics in web documents in two ways, using the `img` element or the `background-image` property.
- The `img` element inserts a graphic in the structure of the document.
- With the `background-image` property it is possible to use a graphic in the background of any element on a web document.

# img

- CSS properties commonly used with images.
  - Borders — border properties can be used to frame an image.
  - Paddings — padding can be used to define the space between a border and an image.
  - Margins — margins can be used to set the distance between the image and other page content.
  - Float — the float property can be used to move the image to the right or left, side-by-side with the surrounding content.



# Background Images

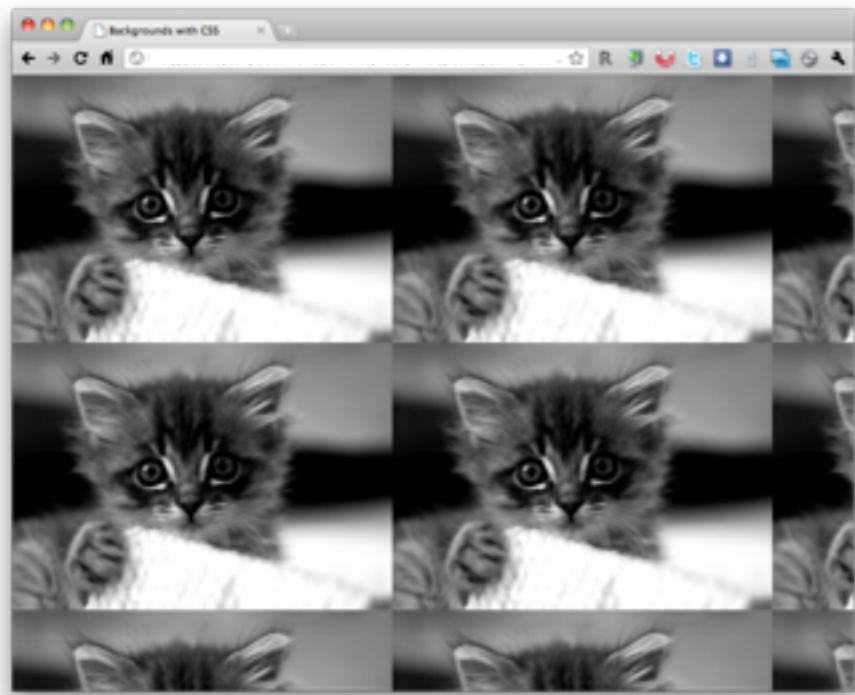
- The `background-image` property adds a graphic to the background of an element.

```
body {  
  background-image: url("media/picture.jpg");  
}
```

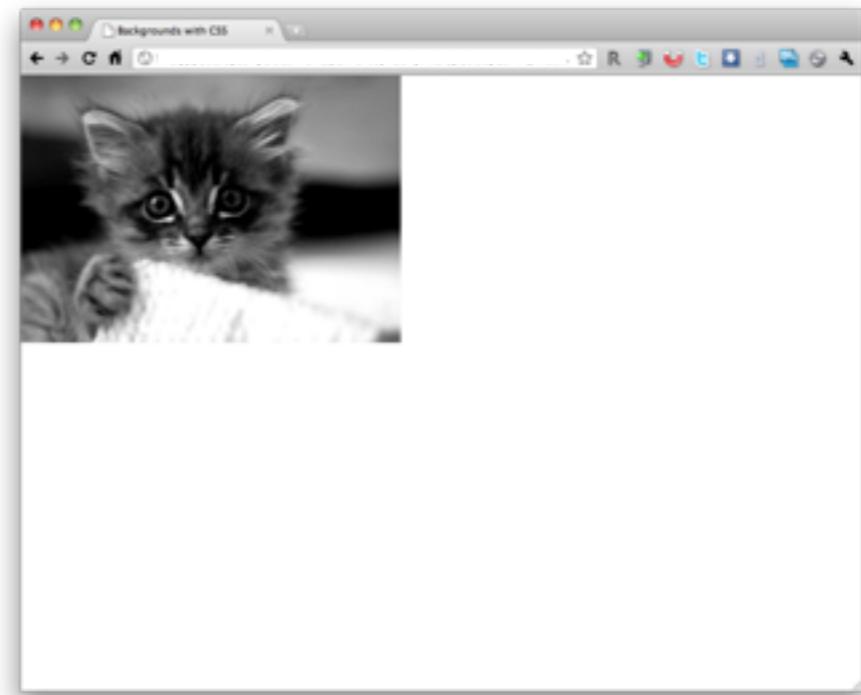
- Absolute or relative URLs can be used.
- It is possible to control both image repetition and image position.

# Background Repetition

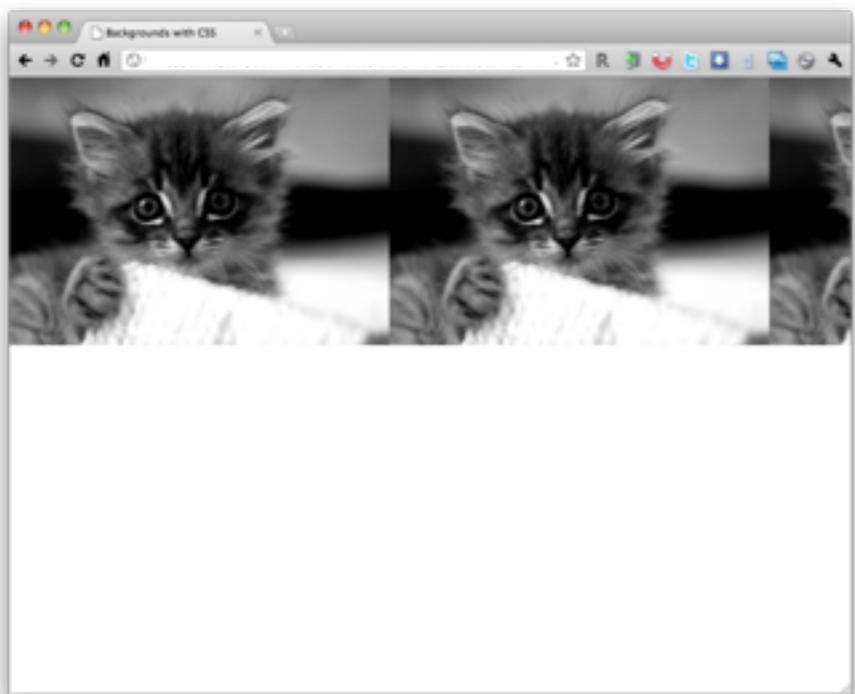
- With the property `background-repeat` is possible to control how the background image is repeated in the available space.
- By default the background graphic is repeated until all the available space is filled.
- Possible values are: `repeat` (the default), `no-repeat`, `repeat-x` and `repeat-y`.



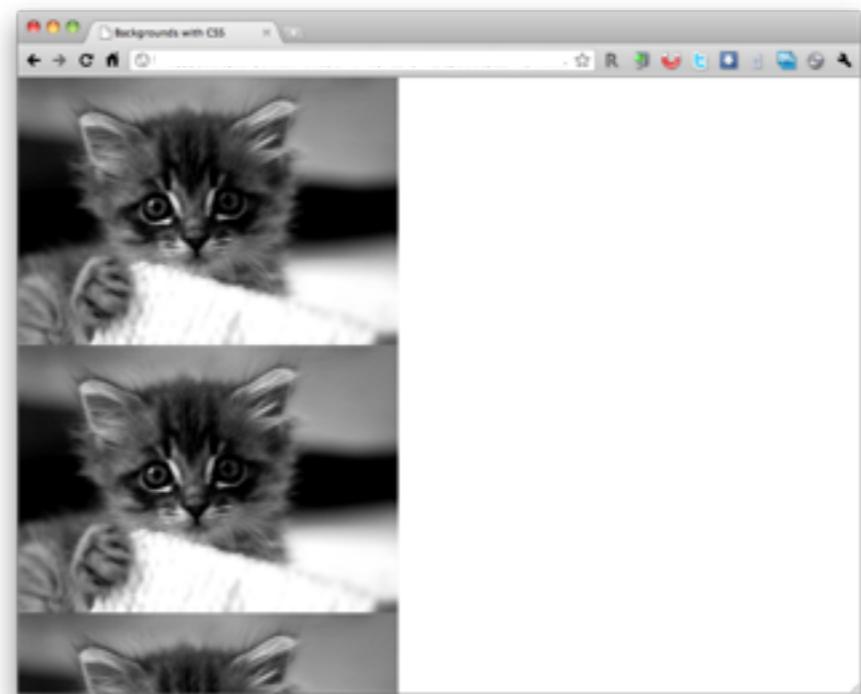
repeat



no-repeat



repeat-x



repeat-y

# Background Position

- With the `background-position` property is possible to define where the graphic should be placed.
- Background position can be defined in three ways: **keywords**, **exact values**, **percentages**.
- It is possible to control the vertical position and the horizontal position.

# Position Values

- **Keywords:** top, bottom, left, right, center.
- **Values:** background position is specified by the distance from the top left corner of the container.
- **Percentages:** background position is proportional to the width and height of the container.



```
body {  
    background-image: url("cat.jpg");  
    background-repeat: no-repeat;  
    background-position: center center;  
}
```



```
body {  
    background-image: url("cat.jpg");  
    background-repeat: no-repeat;  
    background-position: 20px 20px;  
}
```



```
body {  
    background-image: url("cat.jpg");  
    background-repeat: no-repeat;  
    background-position: 20% 20%;  
}
```

# Background Shorthand

- Instead of using three different properties to control background images, the background property shorthand can be used.
- Property **background**, followed by the values for **background-image**, **background-position** and **background-repeat**.
- No need to specify all property values.

```
background-image: url("cat.jpg");  
background-repeat: no-repeat;  
background-position: center center; } background: url("cat.jpg") center center no-repeat;
```

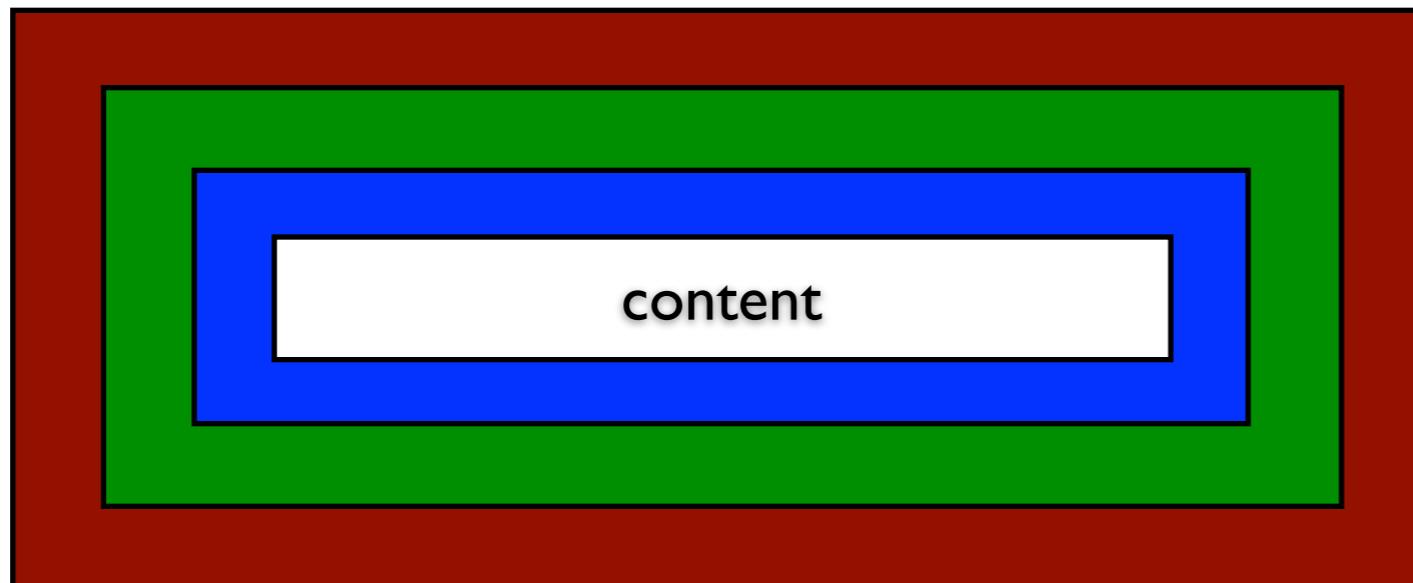
# Finding Images

- <http://search.creativecommons.org>
- <http://www.morguefile.com>
- <http://www.sxc.hu>
- <http://openphoto.net>

# CSS Box Model

# CSS Box Model

- The CSS box model describes the rectangular boxes that are generated for elements in the HTML document.
- Each box has a content and optional surrounding padding, border and margin areas.



padding  
border  
margin

# Border

- The **border** properties specify the width, color, and style of the border area of a box.
- **border-width**: specifies the width of the border area.
- **border-color**: sets the color of the border.
- **border-style**: specifies the line style of a box's border (solid, double, dashed, etc).
- The **border** property is a shorthand for setting the same width, color and style for all four borders of a box. Other shorthands: **border-top**, **border-left**, **border-right**, **border-bottom**.

```
h1 { border-bottom: 1px solid red; }
```

```
strong { border: 1px dashed yellow; }
```

**solid red 3 pixels border**

**:dashed left and right side only green 5 pixels border:**

**dotted bottom side black 2 pixels border**

.....

# Padding

- Padding properties specify the width of the padding area of a box. The padding shorthand property sets the padding for all four sides.
- It is possible to independently control the padding for each side with: padding-top, padding-left, padding-right, padding-bottom.

```
h1 {  
    padding: 10px;  
    padding-bottom: 20px;  
}
```

all around 20 pixels padding

all around 0 pixels padding

top and bottom 10 pixels, left and right 30 pixels

# Margin

- Margin properties specify the width of the margin area of a box. The margin shorthand property sets the margin for all four sides.
- It is possible to independently control the margin for each side with: margin-top, margin-left, margin-right, margin-bottom.

```
h1 {  
    margin-top: 10px;  
    margin-bottom: 50px;  
}
```

# Background

- The **background** of an element can be specified as either a color or an image.
- The background property shorthand refers to the background of the content, padding and border areas.

```
background: red;
```

```
background: url("picture.png") center repeat-y;
```

# Positioning

# CSS Positioning

- A box may be laid out according to three positioning schemes: normal flow, float, absolute positioning.
- Normal flow is the default scheme used for positioning. Boxes are laid out starting at the top and flow from left to right.
- In the float model, a box is laid out according to the normal flow and then taken out of the flow and shifted to left or right. Other content may flow along the side of a float.
- In the absolute positioning model, a box is removed from the normal flow entirely and assigned a new position.

# position and float

- The position and float properties determine which positioning algorithm is used to calculate the position of a box.

# position

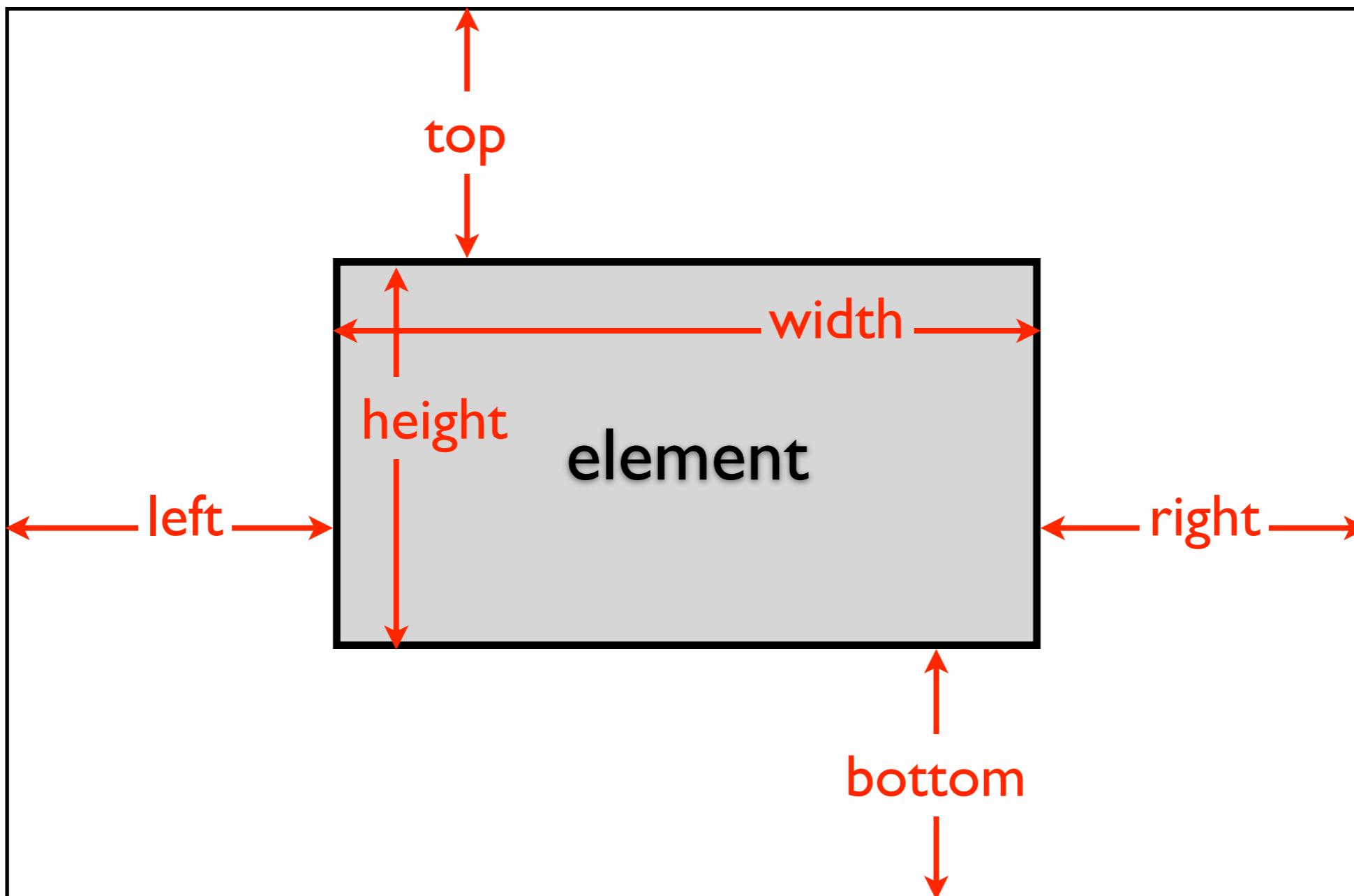
- The position property may have four values:
  - static: the box is positioned according to the normal flow (default value).
  - relative: the box's position is calculated according to its position in the normal flow.
  - absolute: the box's position is specified with the top, right, bottom and left properties.
  - fixed: the box's position is calculated according to the absolute model, but in addition, the box is fixed with respect to the viewport (i.e. browser window).

# Positioning Properties

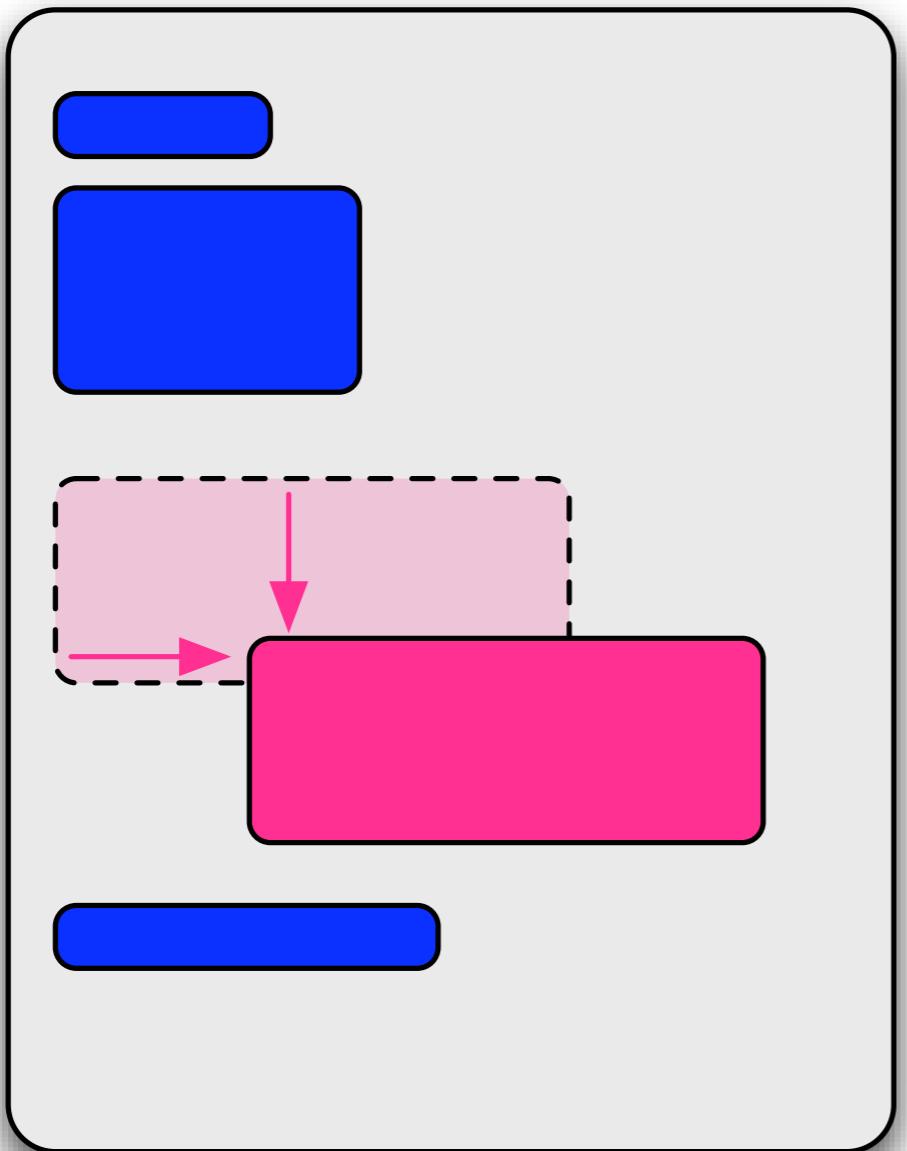
- The `position` property is used together with four box offset properties: top, right, bottom, and left.
- When laying out elements, it is also possible to specify an element's width and height.
- Units may be fixed or relative.

```
#logo {  
  position: absolute;  
  top: 0px;  
  left: 0px;  
}
```

```
#logo {  
  position: relative;  
  bottom: 50px;  
}
```

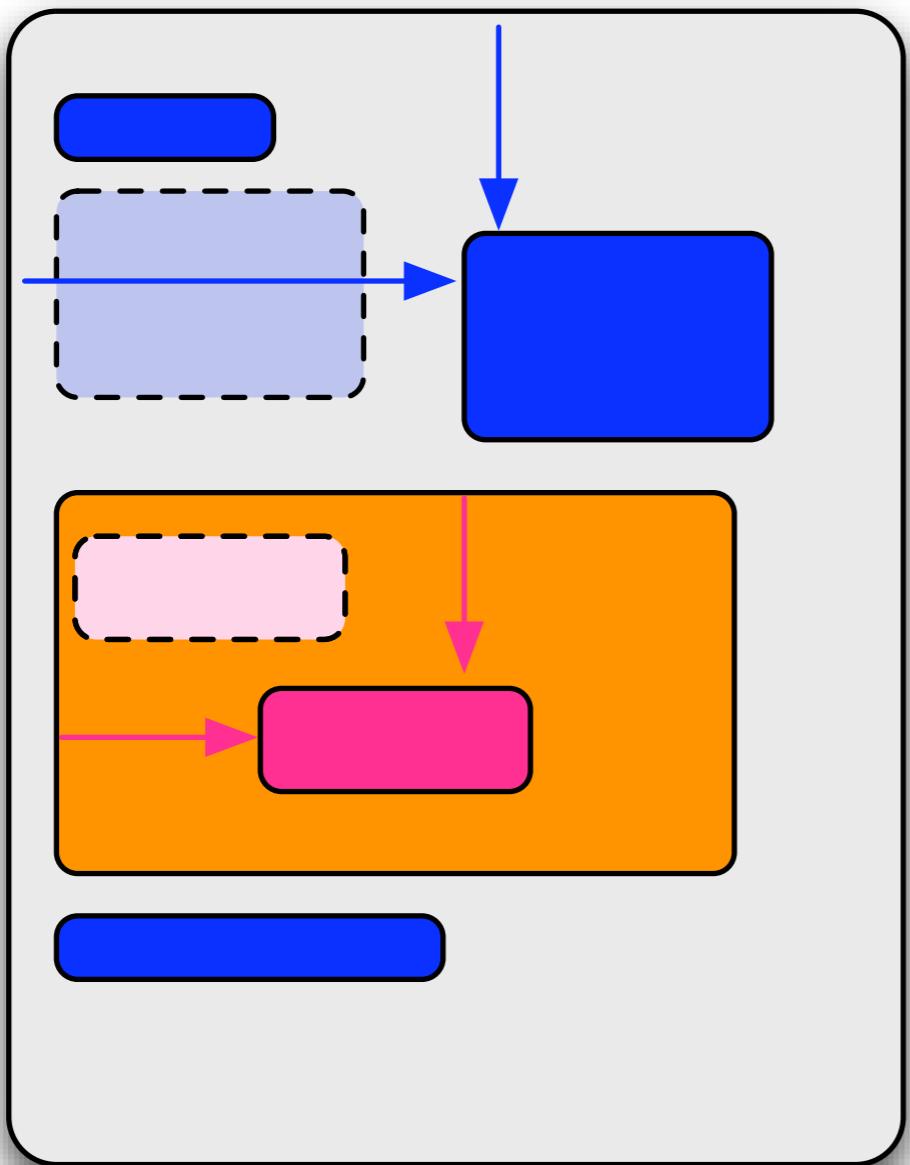


# Relative Positioning



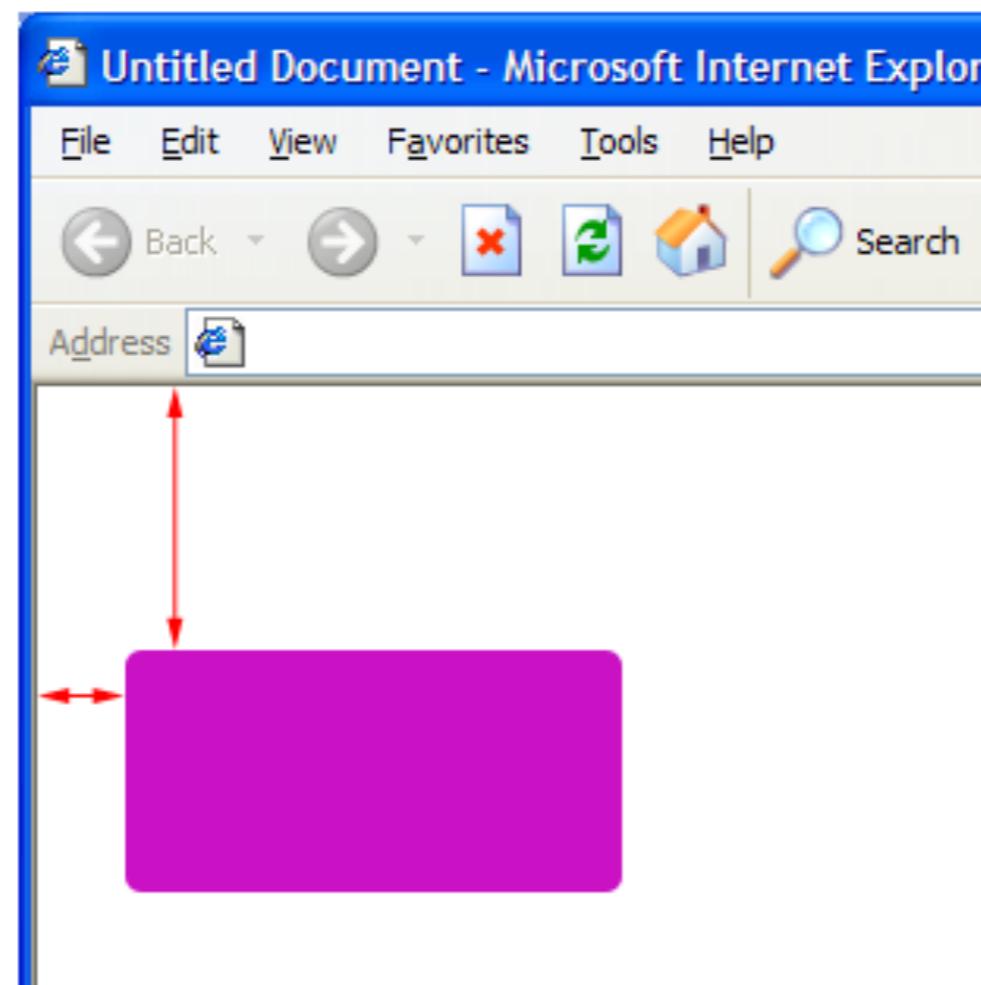
```
#id {  
  position: relative;  
  top: 50px;  
  left: 50px;  
}
```

# Absolute Positioning



```
#id {  
  position: absolute;  
  top: 150px;  
  left: 300px;  
}
```

# Fixed Positioning



# float

- The float property specifies whether a box should float outside its normal flow. There are three possible values:
  - none: the default value, the element is not floated.
  - left: the element is floated to the left and content flows on the right side of the box.
  - right: the element is floated to the right and content flows on the left side of the box.



```
img {  
  float: none;  
}
```

```
img {  
  float: left;  
}
```

```
img {  
  float: right;  
}
```

# z-index

- When elements are removed from their normal flow, e.g. with absolute positioning, they can be stacked inside other elements.
- With the z-index property it is possible to control how elements are stacked (i.e. projected on the z-axis).
- The higher the value, the closer to the user an element is.

```
#Logo {  
background: url(logo.png) top left no-repeat;  
z-index: 10;  
}
```

```
#logo2 {  
background: url(logo2.png) top left no-repeat;  
z-index: 20;  
}
```

# Web Page Layouts

- Web page layouts fall into one of two types, fixed width or liquid.
- Fixed width layouts are independent of the user's viewport size and offer greater control to the developers. The majority of web sites are fixed width. 960px width is a common setting.
- Liquid layouts adjust to the browser's width and height. They are more challenging to the developer. Pure liquid layout are rarely used in practice.

# CSS Techniques

# CSS Reset

# CSS Reset

- Each browser sets different default properties for each element.  
How to deal with this?
- Reset all properties to make all browsers equal. Can be made manually or using an existing library (e.g. YUI CSS Reset).

```
<link rel="stylesheet" type="text/css"  
href="http://yui.yahooapis.com/3.9.1/build/cssreset/cssreset-min.css">
```

# Horizontal Menu

# Horizontal Menu

```
<div id="navcontainer">
<ul id="navlist">
  <li id="active"><a href="#" id="current">Item one</a></li>
  <li><a href="#">Item two</a></li>
  <li><a href="#">Item three</a></li>
  <li><a href="#">Item four</a></li>
  <li><a href="#">Item five</a></li>
</ul>
</div>
```

# Horizontal Menu

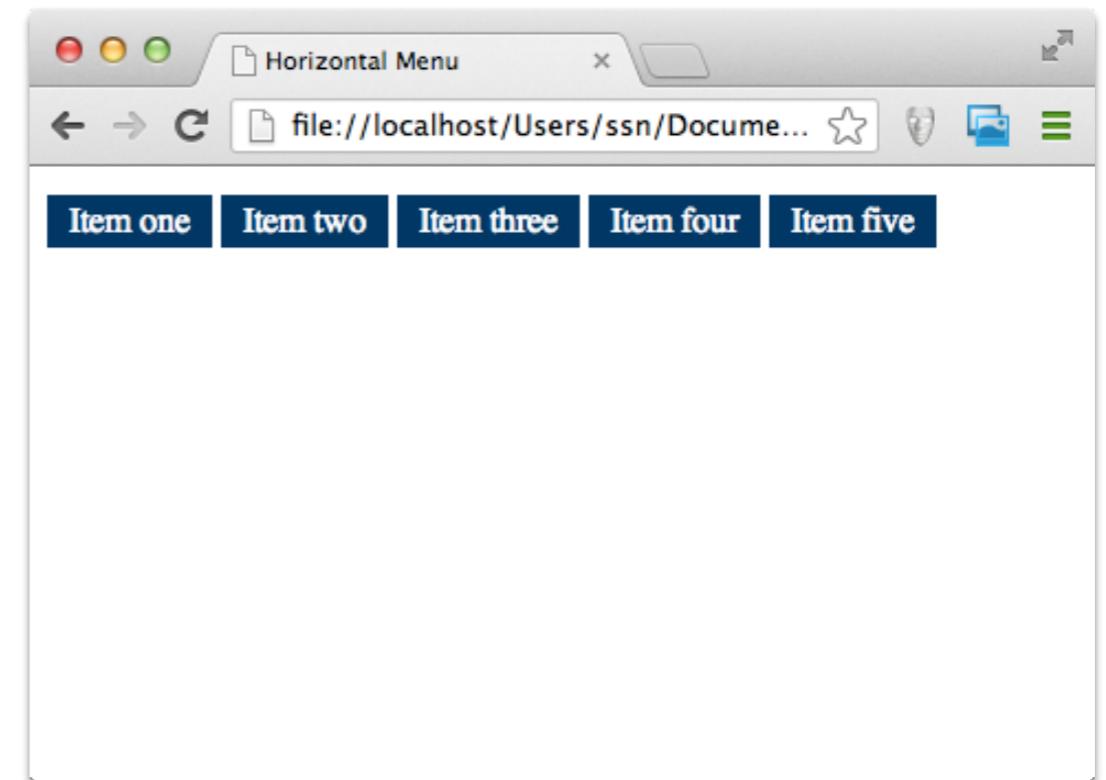
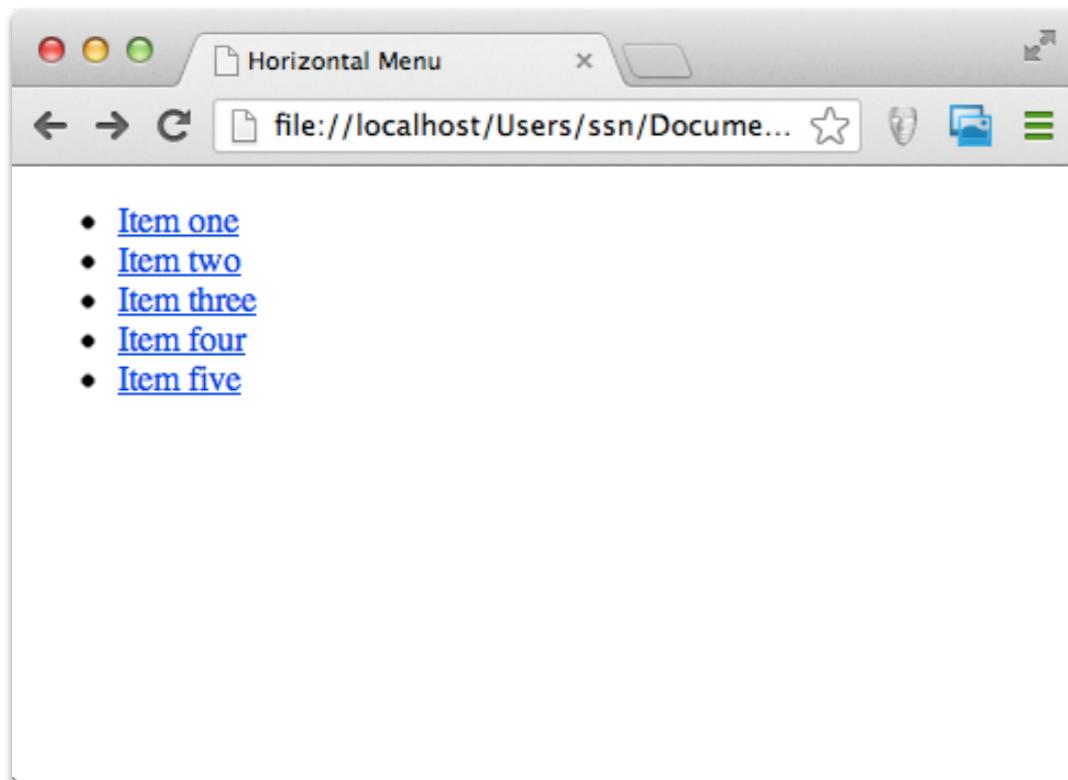
```
ul#navlist {  
    margin-left: 0;                                #navlist a:link, #navlist a:visited {  
    padding-left: 0;                                color: #fff;  
    white-space: nowrap;                            background-color: #036;  
}  
                                         text-decoration: none;  
                                         }  
  
#navlist li {  
    display: inline;                                #navlist a:hover {  
    list-style-type: none;                            color: #fff;  
}  
                                         background-color: #369;  
                                         text-decoration: none;  
                                         }  
  
#navlist a { padding: 3px 10px; }
```

from Max Design

<http://css.maxdesign.com.au/listamatic/horizontal02.htm>

# Horizontal Menu

## Before & After



# Active Item

# Active Item

Standard approach requires a different  
HTML pages per option.

```
<ul>
  <li class="active">Home</li>
  <li>Company</li>
  <li>Contacts</li>
</ul>
```

Home page

```
<ul>
  <li>Home</li>
  <li class="active">Company</li>
  <li>Contacts</li>
</ul>
```

Company page

```
li { color: black; }
li.active { color: red; }
```

# Active Item

With CSS it is possible to have a fixed HTML code.

use a different class for body in each page

```
<body class="company">
```

...

```
<ul>
```

```
  <li class="home">Home</li>
```

```
  <li class="company">Company</li>
```

```
  <li class="contacts">Contacts</li>
```

```
</ul>
```

HTML file

```
li { color: black; }
```

```
body.home li.home,  
body.company li.company,  
body.contacts li.contacts {  
  color: red;  
}
```

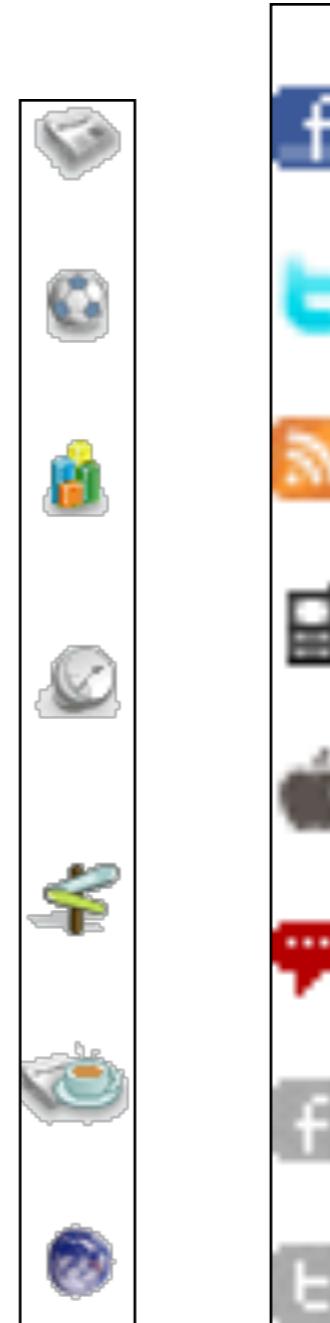
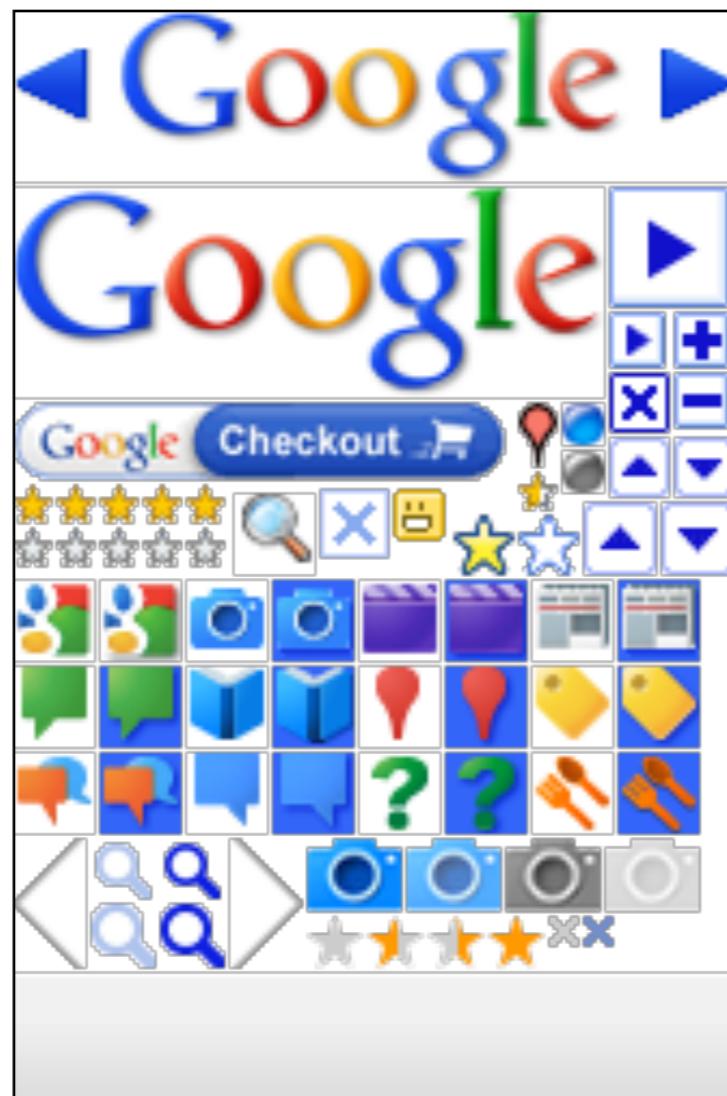
CSS file

# CSS Sprites

# CSS Sprites

- Group multiple images into one single composite image and display using CSS background positioning.
- This technique can save a significant amount of HTTP requests.
- Can also improves the responsiveness of user interfaces by preloading all images.

# Sprites Examples



# CSS Sprites Workings



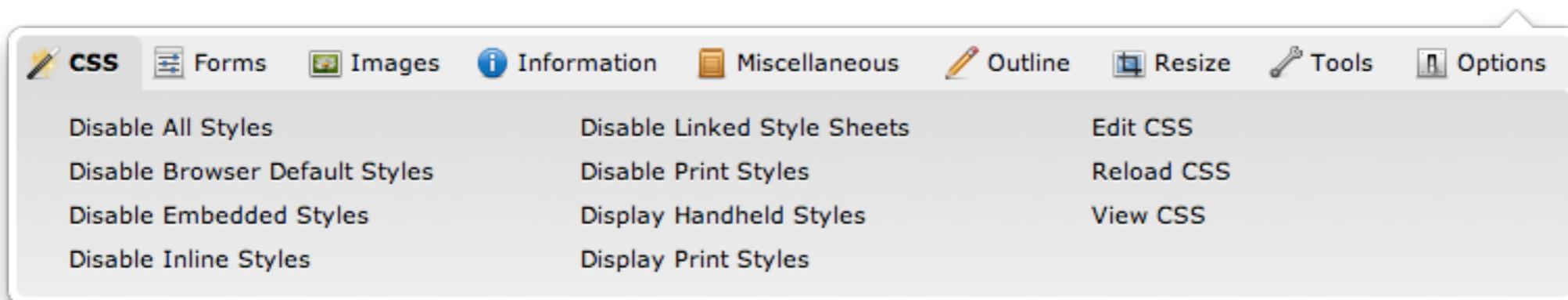
```
a {  
background: url("sprite.png")  
0px 0px no-repeat;  
}
```

```
a:hover {  
background-position: 0px -100px;  
}
```

# Tools

# Web Developer

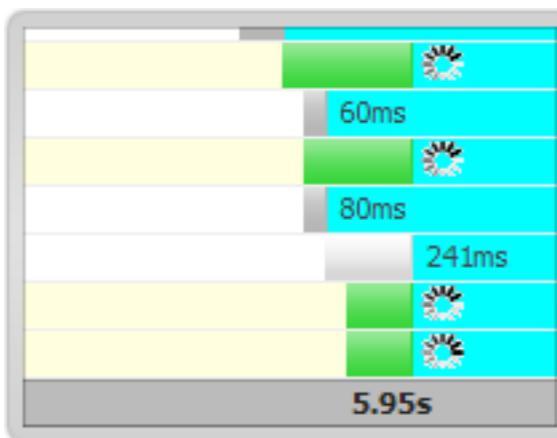
- "The Web Developer extension adds various web developers tools to a browser." [for Firefox and Chrome]



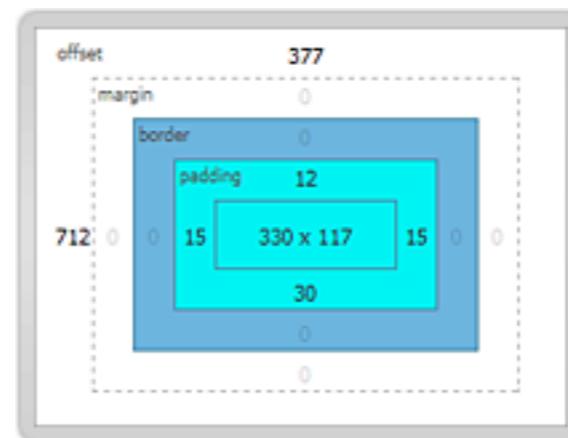
<http://chrispederick.com/work/web-developer/>

# Firebug

- "Firebug integrates with Firefox to put a wealth of web development tools at your fingertips while you browse. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page."



Network activity monitoring

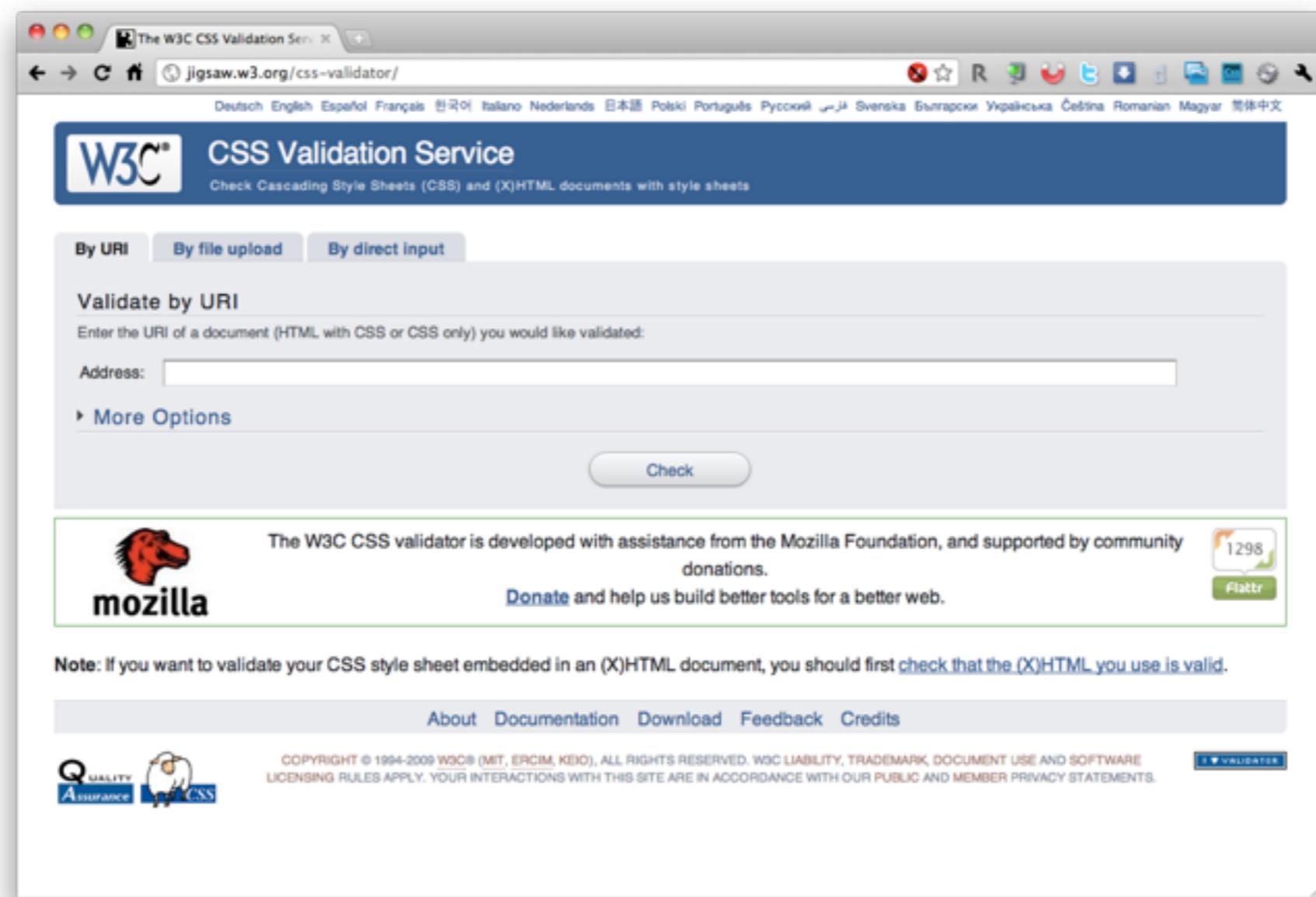


CSS layout

```
---  
id="articleheader">  
id="secondary" class="group"  
<div id="contentarea" class="group">  
  <div id="content" class="content">  
    <h2 class="heading">Wh...  
    <div class="article">  
    </div>  
  <div id="links">  
    <h2 class="heading">Li...
```

HTML inspection

# CSS Validation



<http://jigsaw.w3.org/css-validator/>

# CSS3

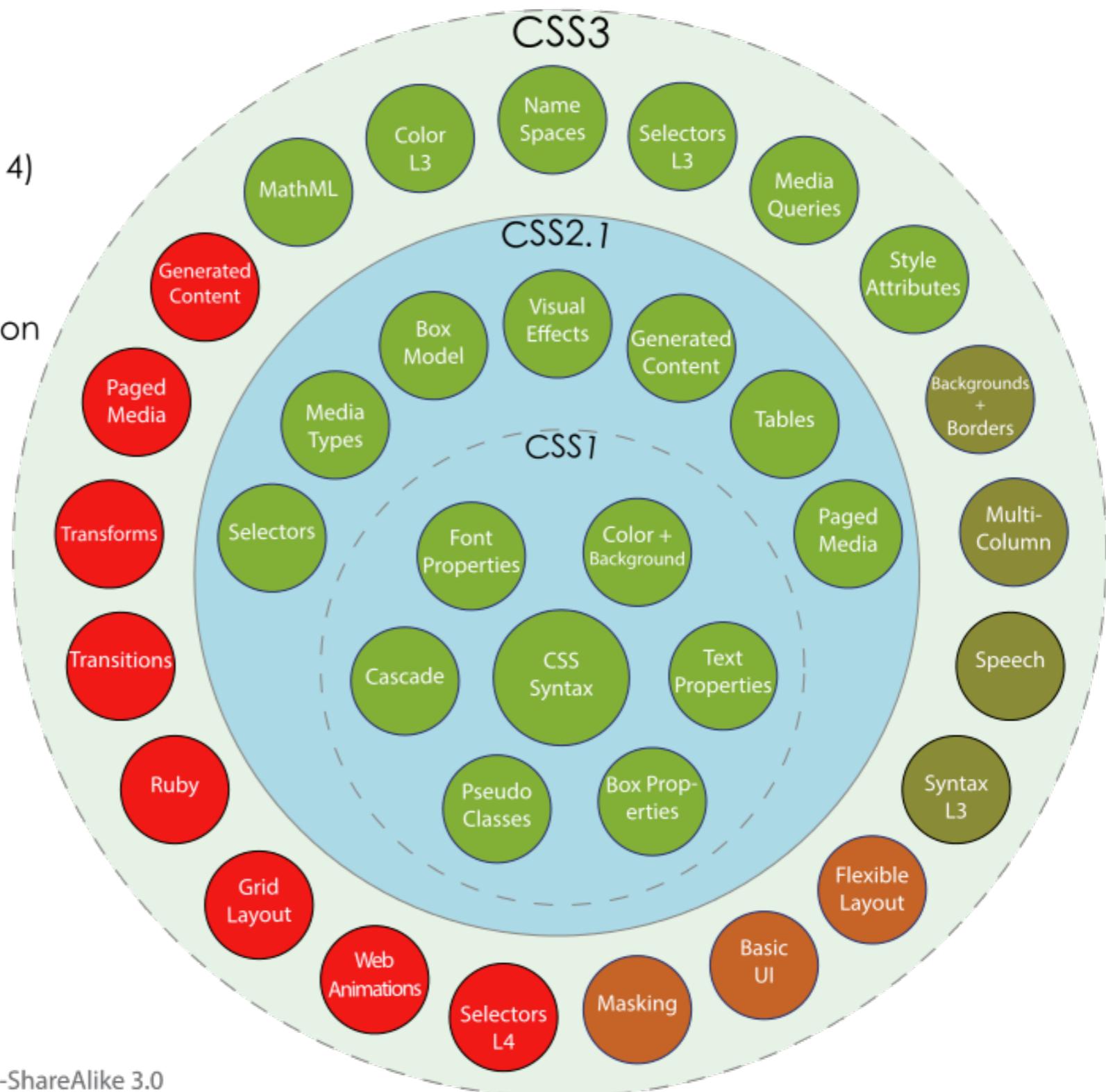
# CSS3

- Work on CSS3 started in 1999.
- Contrary to previous versions, CSS3 is organized in several independent specifications, called modules.
- With this decision, the W3C expects to achieve a more flexible and timely evolution of the specification as a whole.
- Due to this modularization, different modules have different stability and statuses.
- There are more than 50 modules in development. Nine are currently recommendations or candidate recommendations from the W3C.

# CSS3

Taxonomy & Status (October 2014)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Obsolete or inactive



By Sergey Mavrody 2011-14 | CC Attribution-ShareAlike 3.0

# Browser Prefixes

- Browser vendors test experimental features by using browser prefixes. E.g.: -webkit-, -moz-, -o-, -ms-, -chrome-.
- These dash-limited prefixes are added to the beginning of CSS properties to identify work-in-progress features.
- Example: -webkit-border-radius

# CSS3 Transitions

# CSS Transitions

- CSS Transitions allow property changes in CSS values to occur smoothly over a specified duration.
- Transitions are added to the normal state of the element, and triggered when the element changes state (e.g. hover, focus, active, etc).
- Transitions can be applied to many properties, including: background, color, width, opacity, position, and font-size.
- See: <http://www.w3.org/TR/css3-transitions/>

# Transition Properties

- **transition-property**: specifies the property to which the transition is applied to.
- **transition-duration**: specifies the length of time that a transition takes.
- **transition-delay**: defines when the transition will start.
- **transition-timing-function**: defines how the intermediate values of a transition are calculated. Possible values include: ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier.
- **transition**: **<property> <duration> <timing-function> <delay>**

# Transitions Example

HTML    `<a href="#" class="foo">Test</a>`

CSS    `a.foo {  
    background: red;  
    -webkit-transition: background 0.5s ease 0.1s;  
    -o-transition: background 0.5s ease 0.1s;  
    -moz-transition: background 0.5s ease 0.1s;  
}  
  
a.foo:hover {  
    background: darkred;  
}`

# CSS Color

# CSS Color

- The CSS Color module specifies the color values, units and properties for defining the foreground color and opacity of elements.
- The **opacity** property and the **RGBA** color values are two widely supported new features.
- See: <http://www.w3.org/TR/css3-color/>

# Opacity

- The **opacity** property controls the transparency of an element.
- The value of the property defines the degree of transparency to apply to the element, from fully opaque (1.0) to fully transparent (0.0).

# RGBA

- The **RGBA** color model is an extension to the RGB model to include "alpha" to define the opacity of a color.
- Fully opaque red — `rgba(255,0,0,1)`.
- Fully transparent red — `rgba(255,0,0,0)`.

# CSS Text

# CSS Text

- The CSS Text module defines properties for text manipulation, covering line breaking, justification and alignment, white space handling, text decoration and text transformation.
- See: <http://www.w3.org/TR/css3-text/>

# Text Shadow

- The text shadow property accepts a comma-separated list of shadow effects to be applied to the text of the element.
- **Shadow effect:** inset? [<h-offset> <v-offset> <blur> <color>?]
  - **inset:** if present, uses an outer shadow instead of an inner shadow.
  - **h-offset:** horizontal offset of the shadow.
  - **v-offset:** vertical offset.
  - **blur:** blur radius.
  - **color:** color of the shadow.
- Note: originally proposed in CSS2.

# CSS Backgrounds and Borders

# CSS Backgrounds and Borders

- The main extensions compared to CSS2 are borders consisting of images, boxes with multiple backgrounds, boxes with rounded corners and boxes with shadows.
- See: <http://www.w3.org/TR/css3-background/>

# Rounded Corners

- The **border-radius** property defines the radii of a quarter ellipse that defines the shape of the corner of the outer border edge.
- `border-radius: <horizontal radius> [<vertical radius>]`
- Each border can be controlled separately with **border-top-right-radius**, **border-bottom-right-radius**, **border-bottom-left-radius**, **border-top-left-radius**.

# Multiple Backgrounds

- The background of a box can have multiple layers in CSS3.
- The number of layers is determined by the number of comma-separated values in the `background-image` property.
- Each image is sized and positioned according to the corresponding images in the other background properties, also organized in a comma-separated list.

# CSS 2D Transforms

# CSS 2D Transforms

- Using CSS 2D Transforms, elements can be translated, rotated and scaled in two dimensional space.
- Using the `transform` property, a two-dimensional transformation is applied to the coordinate system. This property contains a list of transform functions, applied in the order provided.
- Possible functions: `scale()`, `translate()`, `rotate()`.
- See: <http://www.w3.org/TR/css3-2d-transforms/>

# CSS Animations

# CSS Animations

- The CSS Animations module describes a way for authors to animate the values of CSS properties over time, using **keyframes**.
- The behavior of these **keyframe** animations can be controlled by specifying their **duration**, **number of repeats**, and **repeating behavior**.
- Properties: **animation-duration**, **animation-timing-function**, **animation-direction**, **animation-delay**, etc.
- See: <http://www.w3.org/TR/css3-animations/>

# Media Queries

# Media Queries

- A media query consists of a media type and zero or more expressions that check for the conditions of particular media features.
- Using media queries, presentations can be tailored to a specific range of output devices without changing the content itself.
- Among the media features that can be used in media queries are **width**, **height**, and **color**.
- See: <http://www.w3.org/TR/css3-mediaqueries/>

# Media Queries Example

```
@media screen and (max-width: 600px) {  
  .foo {  
    color: red;  
  }  
}
```

Only applied  
a maximum

Only applied in screens with a maximum width of 600px.

```
@media screen and (min-width: 600px) and (max-width: 900px) {  
  .foo {  
    color: red;  
  }  
}
```

Only applied in screens with a width between 600px and 900px.

Only applied in screens with a width between 600px and 900px.

# Multi-Column Layout

# Multi-Column Layout

- The CSS Multi-column Layout module introduces features that allow the content of an element to be fitted into columns with a gap and a rule between them.
- It is possible to define: the number and width of columns, column gaps and rules, column breaks, spanning columns, etc.
- See: <http://www.w3.org/TR/css3-multicol/>

Having insurance also changes the way I see my body. When I was uninsured my body belonged only to me. When I was sick, I waited out the sore throat and fever in bed, but my body with benefits is partially the responsibility of others. I have a new doctor, a nice woman in Brooklyn. She tells me that I weigh too much, that I must take better care of myself, smoke fewer cigarettes and eat less salty food. I need to get on a plan, she says. And once I get on that plan, I think, I can get married, and my wife can share in my insurance, and we can have children, and know that they will be protected by the full benefits of success and progress. Health care gives me a future. I talked about this with my mother, who has only a bare minimum of coverage and worries constantly about her health, she said, "Paul, what can you do? I'm one of the poor people in America now. At least I'm not alone." And she's happy for me that I have entered the insured class and can enjoy the blessings of my benefits.

It was no fun, in those uninsured days, to feel that my life wasn't worth enough to save, somehow having a job, working for other people—it makes me more of a full human being, and worth a doctor's time. So I keep my little blue card in my wallet at all times, and when I cross the street I think to myself, if I get hit by a car, it will be okay. I know that doesn't seem fair, and I don't like the equation that it represents, but that's the deal, and it's the only deal out there right now.



Having insurance provides a continuity to the face of life. If you're uninsured and you get hit by a car, you are basically screwed. While you live up you'll have a hard time making a living, and once you're landed you'll have an insurmountable bill as long as a need to pay down for the next several years. But if you're covered, you'll spend some time being tended by doctors and nurses, then you'll return to the life you had before you were hurt, and things will be roughly as they were, and you can forget that anything bad ever happened. That's the present. Now that I have benefits I can plan my life in a way that I tragic life, in the form of a heart attack, or a bullet, or some disease, I won't be left to fend for myself.

Having insurance also changes the way I see my body. When I was uninsured my body belonged only to me. When I was sick, I waited out the sore throat and fever in bed. But my body with benefits is partially the responsibility of others. I have a nice doctor, a nice woman in Brooklyn. She tells me that I weigh too much, that I must take better care of myself, smoke fewer cigarettes and eat less salty food. I need to get on a plan, she says. And once I get on that plan, I think, I can get

That insurance provides a continuity to the face of life. If you're uninsured and you get hit by a car, you are basically screwed. While you live up you'll have a hard time making a living, and once you're landed you'll have an insurmountable bill as long as a need to pay down for the next several years. But if you're covered, you'll spend some time being tended by doctors and nurses, then you'll return to the life you had before you were hurt, and things will be roughly as they were, and you can forget that anything bad ever happened. That's the present. Now that I have benefits I can plan my life in a way that I tragic life, in the form of a heart attack, or a bullet, or some disease, I won't be left to fend for myself.

Having insurance also changes the way I see my body. When I was uninsured my body belonged only to me. When I was sick, I waited out the sore throat and fever in bed. But my body with benefits is partially the responsibility of others. I have a nice doctor, a nice woman in Brooklyn. She tells me that I weigh too much, that I must take better care of myself, smoke fewer cigarettes and eat less salty food. I need to get on a plan, she says. And once I get on that plan, I think, I can get

### normal css box model

```
<div id="entry">
  <p>...</p>
  <p>...</p>
</div>
```

### with CSS3 column-count

```
#entry {
  column-count: 2;
}
```

# Sources

- Cascading Style Sheets 2.1 (2010)  
<http://www.w3.org/TR/CSS2/>
- Opera Web Standards Curriculum  
<http://www.opera.com/developer/wsc/>
- Complete CSS Guide from westciv  
[http://www.westciv.com/style\\_master/academy/css\\_tutorial/](http://www.westciv.com/style_master/academy/css_tutorial/)
- CSS: The Missing Manual, 2nd Ed.  
David Sawyer McFarland, O'Reilly (2009)
- HTML & CSS: The Good Parts  
Ben Henick, O'Reilly (2010)
- CSS3 for Web Designers  
Dan Cederholm, A Book Apart (2010)

# Further Reading

- W3C - Cascading Style Sheets  
<http://www.w3.org/Style/CSS/>
- CSS - Dev.Opera  
<http://dev.opera.com/articles/css/>
- The Elements of Typographic Style Applied to the Web  
<http://webtypography.net>