Mobile Web Appplications Development with HTML5

Lecture 5: CSS3 & Responsive Design

Claudio Riva

Aalto University - Fall 2012

Lecture 5: CSS3 & Responsive Design

Progressive Enhancement & Responsive Design

Media Queries

Position: fixed & Overflow: scroll

Touch Interaction

CSS3

Desktop First

Historically the workflow for web design started with desktop first

- Tools like Photoshop, Fireworks, Dreamweaver
- Printed-page metaphor.
- Optimized for a specific width: 960px
- Optimized for a specific browser(s)
- Optimized for one input method: mouse
- Aim at identical experiences cross-browser and cross-platform

Then Mobile Came

Then Mobile Came

1.2 billion Mobile Web Users (2011)

Then Mobile Came

1.2 billion Mobile Web Users (2011)

20-25% Web users are only mobile (mobile vs desktop)

Then Mobile Came

1.2 billion Mobile Web Users (2011)

20-25% Web users are only mobile (mobile vs desktop)

How to adapt desktop web sites to mobile

Then Mobile Came

1.2 billion Mobile Web Users (2011)

20-25% Web users are only mobile (mobile vs desktop)

How to adapt desktop web sites to mobile

- Do nothing => awful experience

Then Mobile Came

1.2 billion Mobile Web Users (2011)

20-25% Web users are only mobile (mobile vs desktop)

How to adapt desktop web sites to mobile

- Do nothing => awful experience
- Separated Mobile Experience
  - 4 different sites (desktop, tablet, touch, non-touch)
  - Mobile users are often penalized (content, interaction and performance)

Device Categories and UX Factors

Devices

- Desktop/laptops
- Tablets
- Smart phones
- Feature phones

Main factors affect UX

- Screen resolution (size, orientation, dpi)
- User interaction (mouse, keyboard, touch, voice)
- Performance (CPU power, memory capacity, GPU, network)
- Default browser

Desktop/Laptop

- Large screen
- High performance
- High bandwidth
- Multiple browsers with different capabilities
- Mouse and keyboard interaction (but not always)

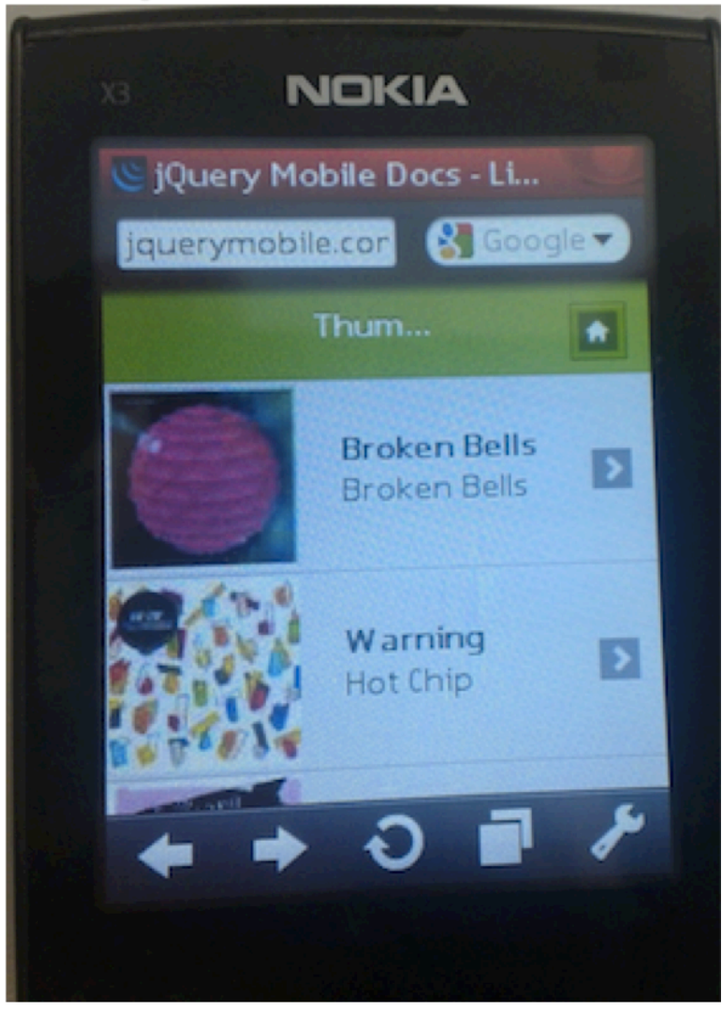
Tablet

- Large screen
- Close to smartphones for user interaction
- Two orientations
- Smooth animations and graphics (often equipped with powerful GPU)
- High to medium performance
- High to medium bandwidth
- Primarily used with the default browser (but not always)
- Touch interaction (but not always)

Smartphones

- Small screen but high-definition
- Handheld device
- Used while on the move
- Two orientations
- Smooth animations and graphics (often equipped with powerful GPU)
- Medium performance
- 3G/Wifi/EDGE/LTE network
- Primarily used with the default browser (but not always)
- Touch interaction

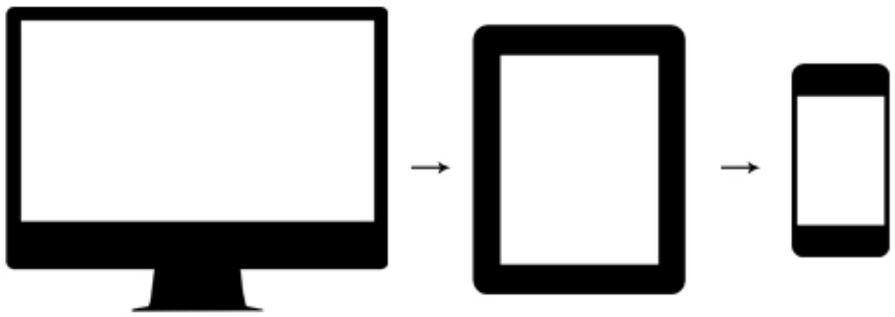
Feature phones



- Small screen
- One or two orientations
- Limited animations and graphics
- Medium-low performance
- 2G/3G bandwidth
- Limited native and proxy browsing
- Touch interaction (but not always)
- Multi-tasking users on the move that require fast and responsive UI
- Task oriented UI
- Online/offline experience

Design for Mobile First

Design for Mobile First

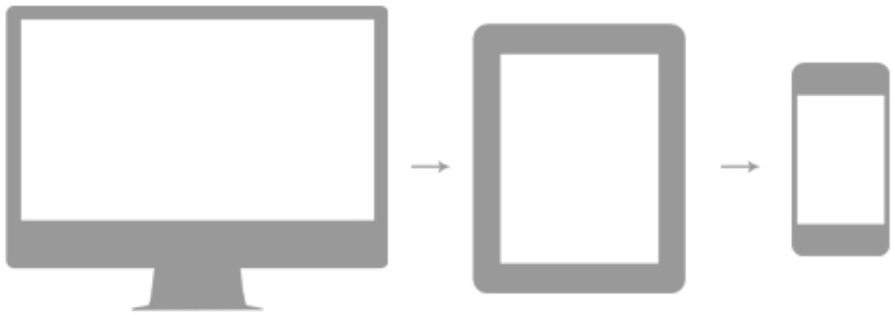


Graceful Degradation

Design for desktop on as many browsers and platforms as possible

Account for possible *degradation* and ensure functionality (e.g. remove content)

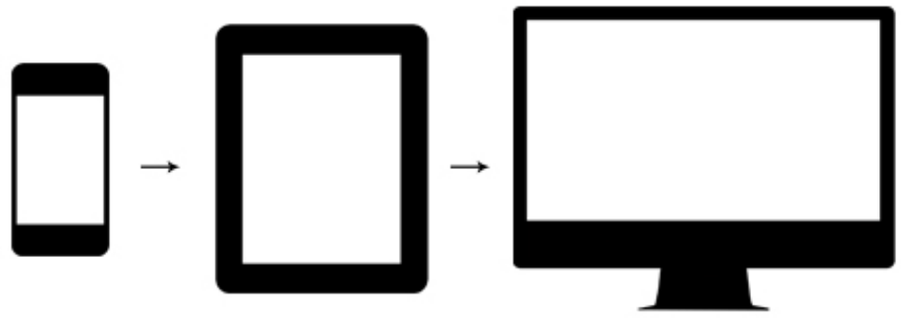
Design for Mobile First



Graceful Degradation

Design for desktop on as many browsers and platforms as possible

Account for possible *degradation* and ensure functionality (e.g. remove content)



Progressive Enhancement

First create a baseline experience that focuses on semantic markup, structure and content

Second start layering presentation and interactivity to provide a richer experience for more capable browsers

Content Parity

One Web Experience

Give people what they want regardless of how they access the Web

The same content is accessible to all devices

Content has structure and semantic

Less-capable browser will only get the content (e.g. jQueryMobile)

Progressive Enhancement More Info

Progressive Enhancement More Info

... is about ensuring the content is available to and usable by anyone regardless of the location, browser, device or capabilities.

Progressive Enhancement More Info

... is about ensuring the content is available to and usable by anyone regardless of the location, browser, device or capabilities.

... is about accessibility because of special needs:

- when watching a website on a phone I'm limited by the screen resolution
- when browsing on a touch device I'm limited by the precision of my fingertips

Progressive Enhancement More Info

... is about ensuring the content is available to and usable by anyone regardless of the location, browser, device or capabilities.

... is about accessibility because of special needs:

- when watching a website on a phone I'm limited by the screen resolution
- when browsing on a touch device I'm limited by the precision of my fingertips

... is about crafting experiences that serve the users by giving access to content without technical restrictions

Progressive Enhancement More Info

... is about ensuring the content is available to and usable by anyone regardless of the location, browser, device or capabilities.

... is about accessibility because of special needs:

- when watching a website on a phone I'm limited by the screen resolution
- when browsing on a touch device I'm limited by the precision of my fingertips

... is about crafting experiences that serve the users by giving access to content without technical restrictions

... is about leveraging the fault tolerance of web browsers

Progressive Enhancement More Info

... is about ensuring the content is available to and usable by anyone regardless of the location, browser, device or capabilities.

... is about accessibility because of special needs:

- when watching a website on a phone I'm limited by the screen resolution
- when browsing on a touch device I'm limited by the precision of my fingertips

... is about crafting experiences that serve the users by giving access to content without technical restrictions

... is about leveraging the fault tolerance of web browsers

... is about applying latest technologies in an intelligent way, layer-upon-layer

Layers of User Experience

Interactivity

Add interactivity to the page through Javascript by listening to events and manipulating the DOM tree

```
$("img").click( function(e) { ... });
```

Audio and visual

The elements of the page can be styled and enriched with CSS and inline images.

```
p {color: red; font-weight: bold; }
```

Semantics

The various elements and attributes in a page provide an additional meaning and context to the text

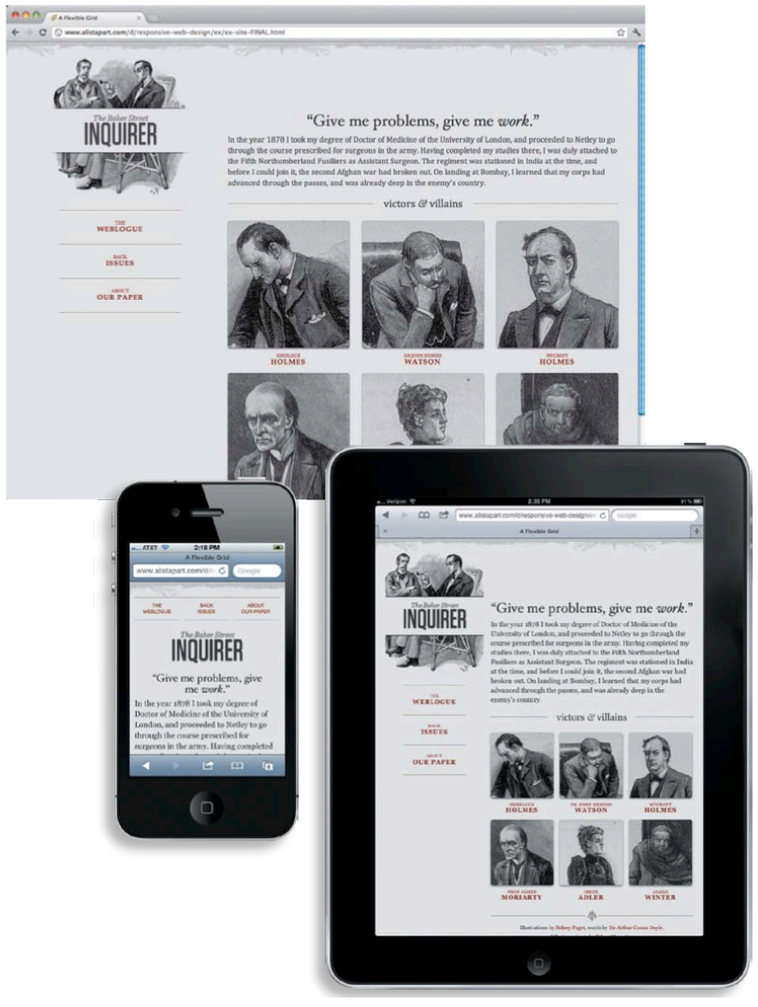
```
<h1><a href\="..."title\="Hello"><img src\="..."alt\="A nice
picture"></a></h1>
```

Text

The basic form of HTML content is text

```
<h1><a href="..." title="Hello"><img src="..." alt="A nice
picture"></a></h1>
```

Responsive Design



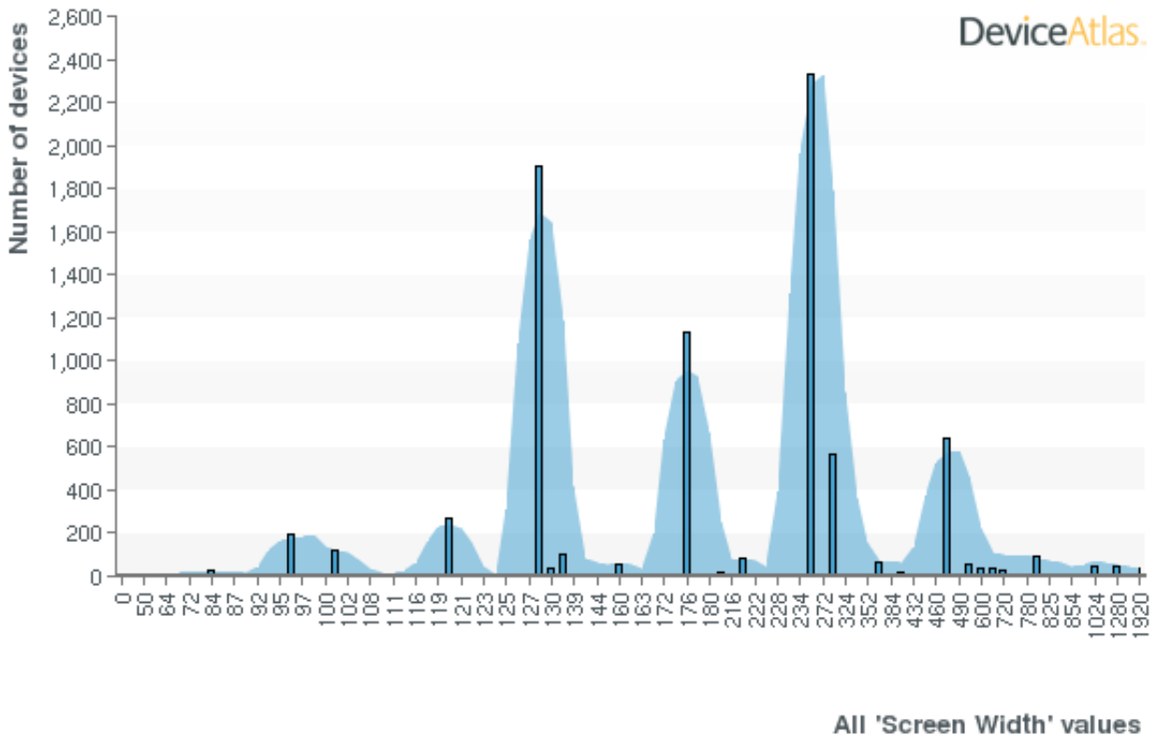
Responsive Web Design by Ethan Marcotte

Demonstrated how to build a responsive design for multiple screen resolutions with:

- Media Queries
- Fluid Grids
- Scalable Images

Some examples: 1 , 2 , and 3

Device Screen Width



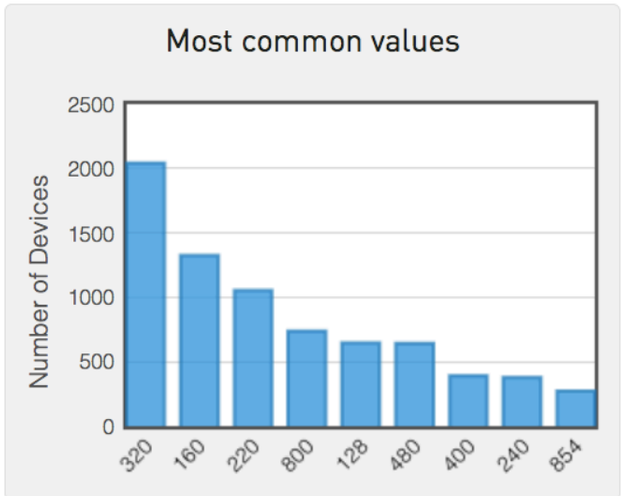
240px (26%)

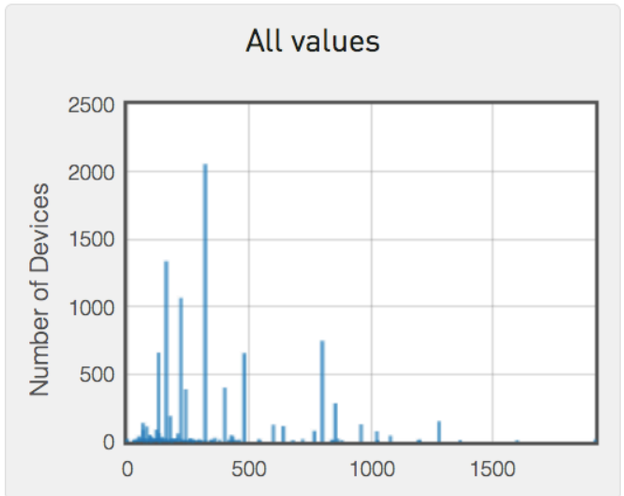
128px (19%)

176px (12%)

480px (10%)

Device Screen Height





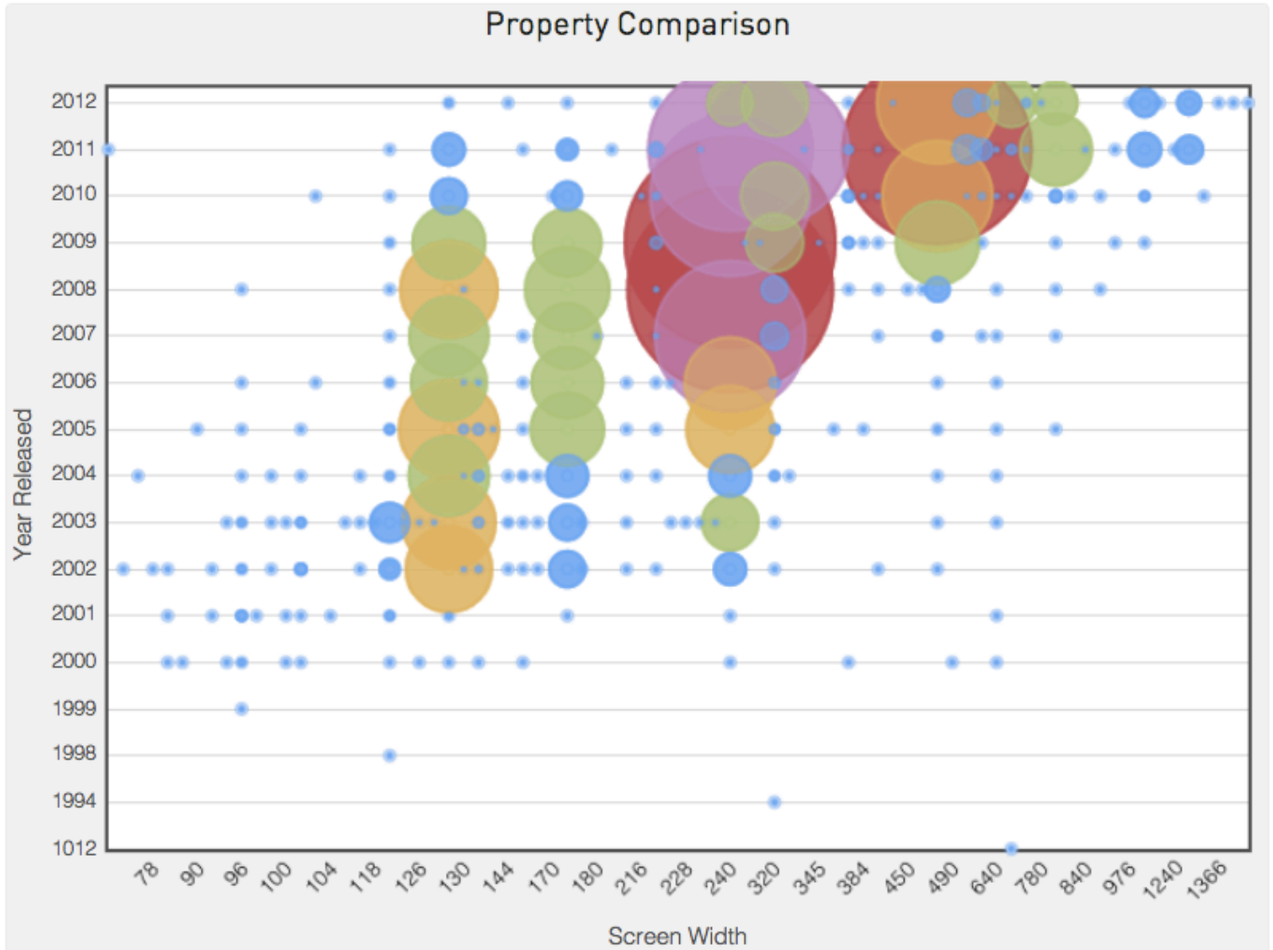
320px (21%)

160px (14%)

220px (11%)

800px (7%)

Device Screen Width over Time



2012 : 128, 144, 176, 220, **240, 320,** 360, 430, **480,** 540, 600, 640, **720,** 768, 780, **800,** 976, 1024, 1080, 1280, 1366, 1920, 2560

Nokia 113 (the smallest)

Nokia 113 (the smallest)

128 x 160 pixels

Nokia 113 (the smallest)

128 x 160 pixels

1.8 inches display (114 ppi)

Nokia 113 (the smallest)

128 x 160 pixels

1.8 inches display (114 ppi)

33€

Nokia 113 (the smallest)

128 x 160 pixels

1.8 inches display (114 ppi)

33€

Does it have a browser ?

Nokia 113 (the smallest)

128 x 160 pixels

1.8 inches display (114 ppi)

33€

Does it have a browser ?

Yes, it does.

Samsung Google Nexus 10 (the biggest)

Samsung Google Nexus 10 (the biggest)

2560 x 1600 pixels

Samsung Google Nexus 10 (the biggest)

2560 x 1600 pixels

10.1 inches display (299 ppi)

Samsung Google Nexus 10 (the biggest)

2560 x 1600 pixels

10.1 inches display (299 ppi)

999,90€

Samsung Google Nexus 10 (the biggest)

2560 x 1600 pixels

10.1 inches display (299 ppi)

999,90€

Does it have a browser ?

Samsung Google Nexus 10 (the biggest)

2560 x 1600 pixels

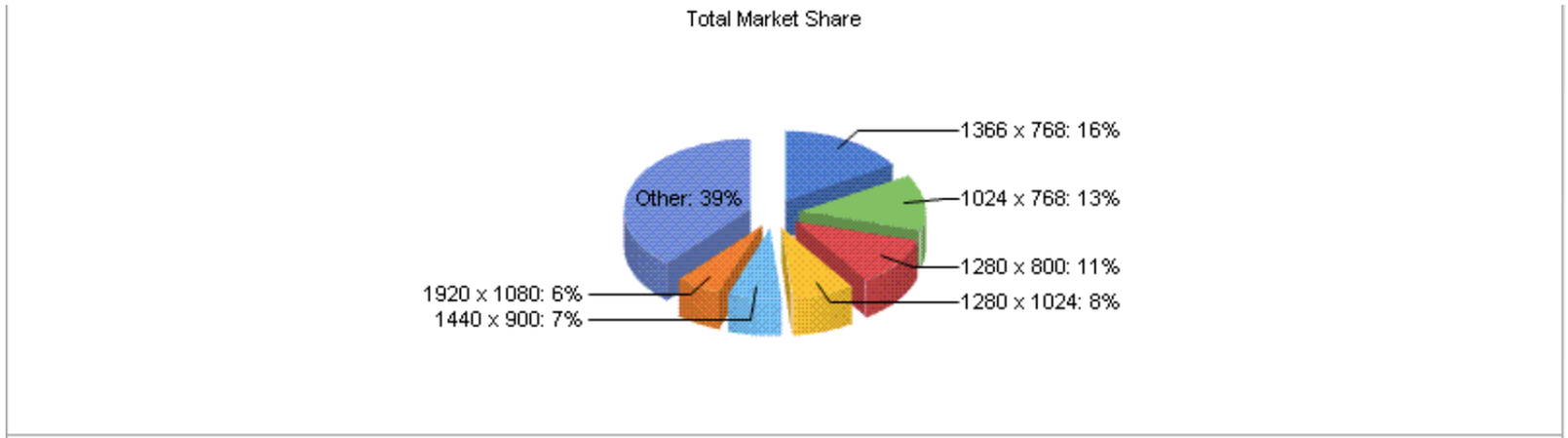
10.1 inches display (299 ppi)

999,90€

Does it have a browser ?

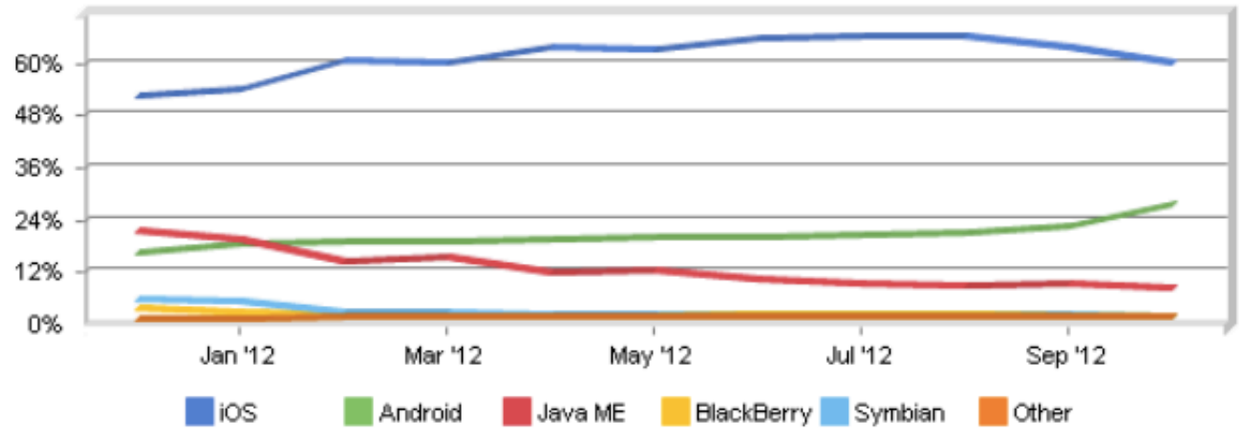
Yes, it does.

Screen Resolution Market Share



RESOLUTION	TOTAL MARKET SHARE
1366 x 768	15.50%
1024 x 768	12.28%
1280 x 800	10.83%
1280 x 1024	7.45%
1440 x 900	6.36%
1920 x 1080	5.66%
768 x 1024	4.86%
320 x 480	4.68%
1600 x 900	3.89%
1680 x 1050	3.74%
1920 x 1200	1.48%
1024 x 600	1.43%

Mobile OS Market Share (based on browser)



Month	iOS	Android	Java ME	BlackBerry	Symbian	Other
December, 2011	52.10%	16.15%	21.27%	3.51%	5.76%	1.20%
January, 2012	53.65%	18.12%	19.19%	2.59%	5.20%	1.24%
February, 2012	60.59%	19.02%	14.16%	2.15%	2.77%	1.32%
March, 2012	59.87%	18.66%	15.10%	2.21%	2.77%	1.40%
April, 2012	63.19%	19.27%	11.83%	2.04%	2.24%	1.42%

Popular screen resolutions for mobile

- **480 x 320** : old iPhones, mid-range Android and Nokia devices
- **320 x 240** : low-end Android and older Nokia Symbian phones
- **800 x 480** : new Nokia and Android devices
- **960 x 640** : iPhone 4
- **1136 x 640** : iPhone 5
- **1024 x 768** : iPad and Android tablets
- **1280 x 768** : Lumia 920

Layouts Options

Fixed-width

- The most common is 960px
- It gives an illusion of controlling the available space
- It gives a bad experience on narrower or wider screens

Layouts Options

Fixed-width

- The most common is 960px
- It gives an illusion of controlling the available space
- It gives a bad experience on narrower or wider screens

Fluid Layouts

- Dimensions are based on percentages, not fixed measurements
- Not enough to scale from small to wide displays.

Layouts Options

Fixed-width

- The most common is 960px
- It gives an illusion of controlling the available space
- It gives a bad experience on narrower or wider screens

Fluid Layouts

- Dimensions are based on percentages, not fixed measurements
- Not enough to scale from small to wide displays.

Elastic layouts

- Dimensions are determined by type size (`em`)
- Good for readability (e.g. by fixing width to 55em)
- Suffers from the fixed-width issues

Layouts Options

Fixed-width

- The most common is 960px
- It gives an illusion of controlling the available space
- It gives a bad experience on narrower or wider screens

Fluid Layouts

- Dimensions are based on percentages, not fixed measurements
- Not enough to scale from small to wide displays.

Elastic layouts

- Dimensions are determined by type size (`em`)
- Good for readability (e.g. by fixing width to 55em)
- Suffers from the fixed-width issues

Hybrid layouts

- A combination of the previous methods

Sizing the fonts

First place where to add *fluidity* to a Web design

- **pixels**
  - not cascading from parent to children
  - issues with zooming, different screen sizes and pixel densities
- **ems**
  - most common method
  - cascading
  - 1 em = current font size (current = context where em is used)
- **percentages**
  - behaves like em
- **rems**
  - behaves like ems but always refers to the root element
  - not well supported on mobile

Font Sizing responsiveFontSizing

```
<body>
  <h1>Lorem Ipsum</h1>
  <div class='container'>
    <div>
      <h2>Chapter 1<span>Lorem ipsum</span></h2>
      <p>Lorem ipsum ...</p>
    </div>
    <div>
      <h2>Chapter 2<span class='smaller'>Aliquam erat</span></h2>
      <p>Aliquam erat ...</p>
    </div>
  </div>
</body>
```

```
body { font-size: 16px; }
h1 { font-size: 2em; } /* 32px */
h2 { font-size: 1.5em; } /* 24px */
span { font-size: 1em; }
span.smaller { font-size: .6666em; } /* 16px / 24px */
```

Grid layout

Break the page in columns based on the content (text, image, video)

- Bring order, consistency and harmony to the page
- Make easier for the user to find content

Grid layout

Break the page in columns based on the content (text, image, video)

- Bring order, consistency and harmony to the page
- Make easier for the user to find content

How to build your own fluid grid:

- Start with a fixed layout
- Break the page in columns of equally fixed size
- Add the structure to your page
- Make images fluid
- Mix fixed and fluid widths for the columns

Fluid Layout responsiveFluidLayout

```
%body
  %header
    %h1 The Greatest Book Ever
  %article
    %h2
      Chapter 1
    %figure
      %img(src="mountain.jpg")
    ...
  %aside
    %section
      %h2 Chapters
        ...
    %section
      %figure
        %img(src="smile.jpg")

    %section
        ...
```

```
article {
  display: table-cell;
}

aside {
  display: table-cell;
  width: 300px;
}

figure img {
  width: 100%;
  max-width: 100%;
}
```

Popular Frameworks (many and more)

Grid-frameworks

Bootstrap

The Semantic Grid System

Less Framework

Foundation 3

HTML5 Boilerplates

Gridless

320 and up

HTML5 Boilerplate

Responsive Design Testing Tool

Media Queries W3C

Query syntax for serving the most appropriate styles based on the device characteristics

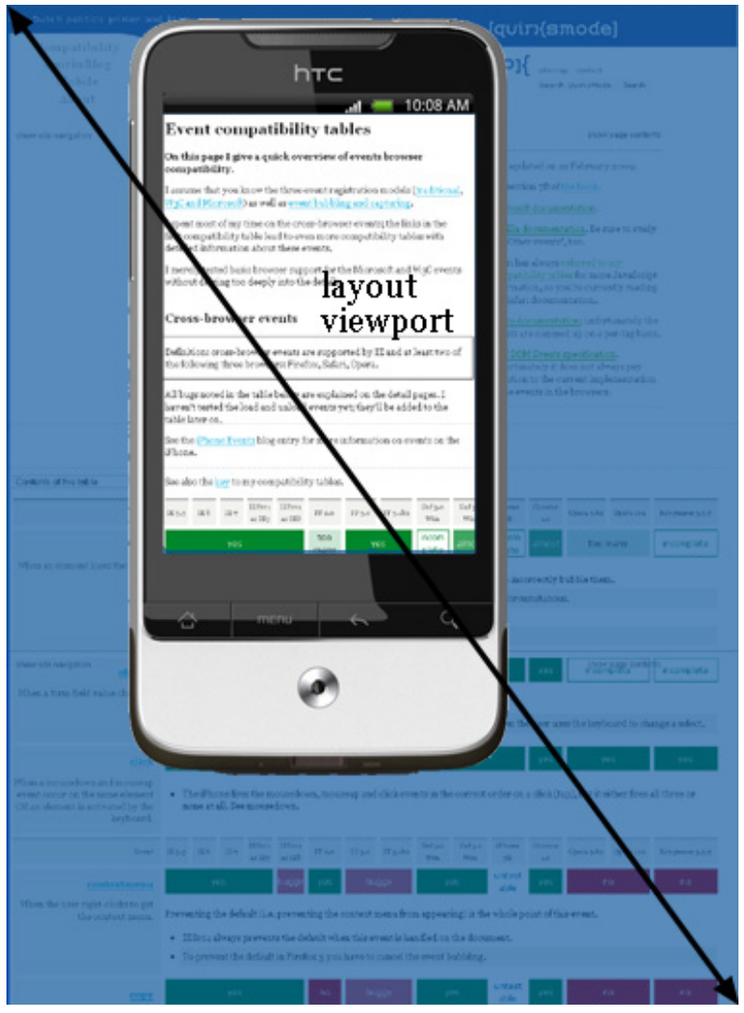
Enables us:

- to create device-independent websites
- to optimize the visitor's experience
- to avoid multiple sites per device (one code base)
- to serve the mobile optimized versions (i.e. smaller images)

What is the Viewport ?

What is the Viewport ?

- The available area for rendering a web page
- Not necessarily equal to the visible area on the device
- User can zoom and pan within the viewport
- Default viewport width:
  - 980px on iPhone
  - 850px on Opera
  - 800px on Android



Device pixel vs CSS pixel

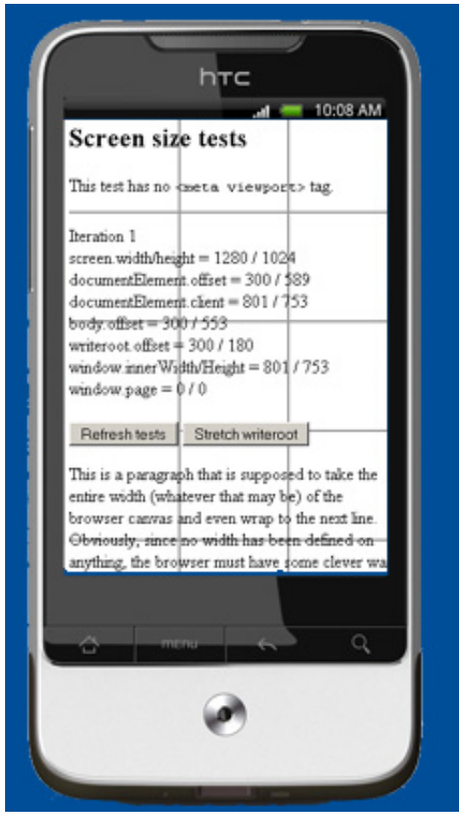
Device pixel : physical pixels of the device (e.g. 1024px wide screen)

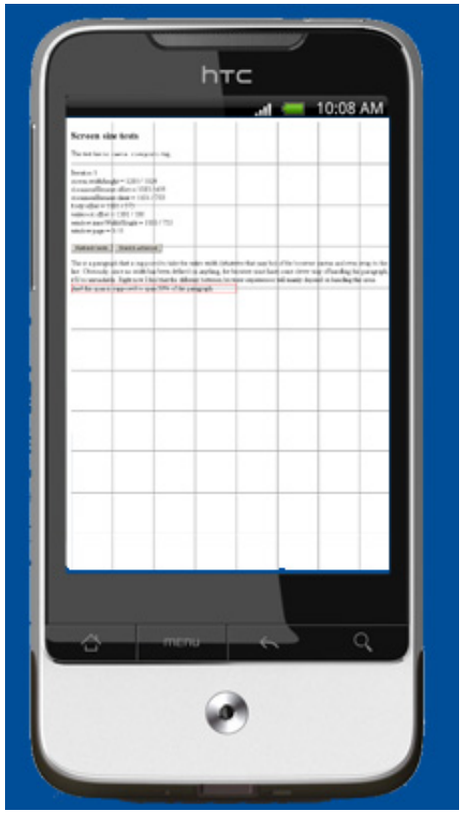
CSS pixel : virtual pixels of the Web page (e.g. 960px or 2000px wide page)

CSS pixels do not necesarilly map to physical pixels

- zooming alters the ratio between physical and virtual pixels
- high-resolution display (e.g. Retina)

Device pixel vs CSS pixel



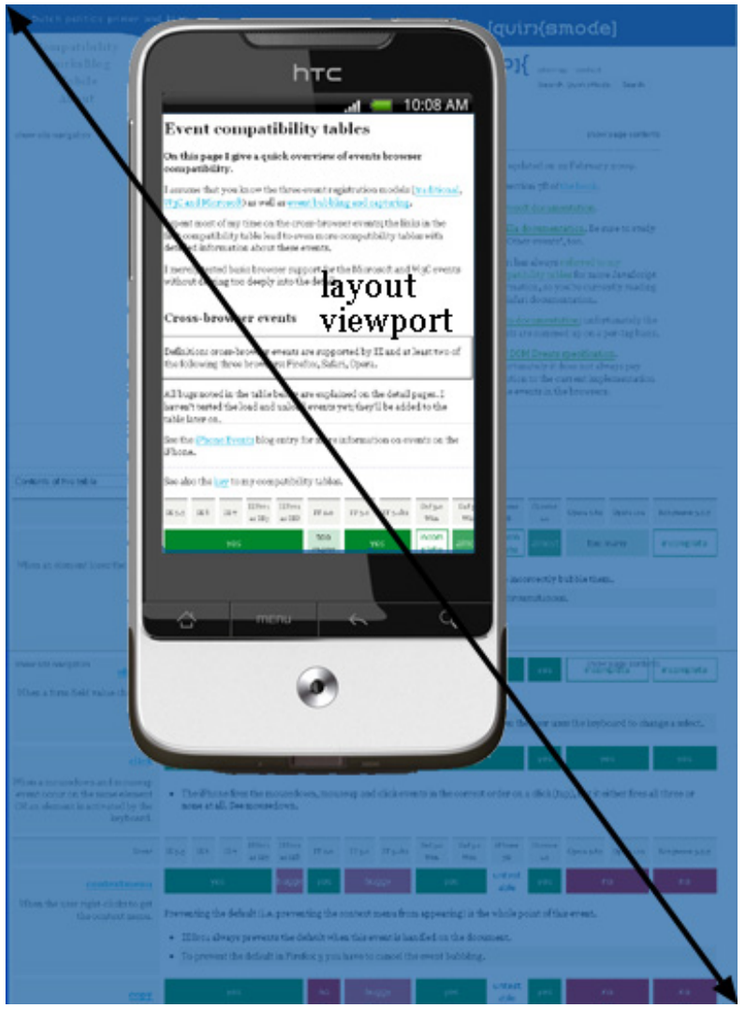


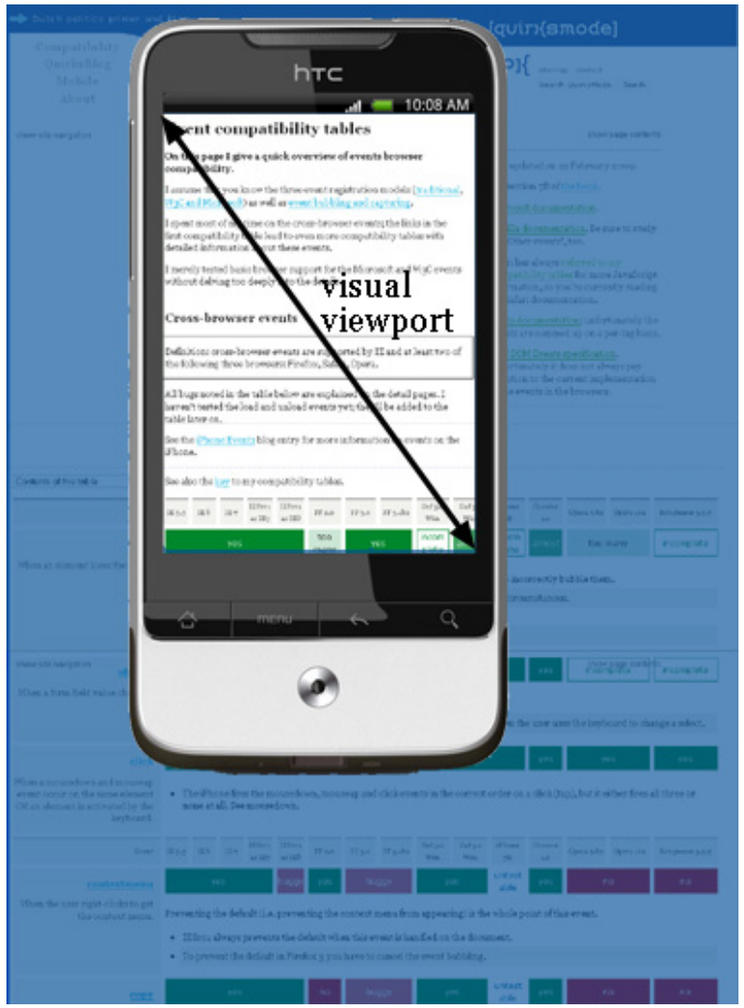
Layout viewport vs visual viewport

Layout viewport : refers to the large image where the page is rendered. It doesn't change in size or shape.

Visual viewport : refers to the smaller frame that we can use to look at the bigger image. With the small frame we can back away, or move closer or change the orientation we use for looking at the large image.

Layout viewport vs visual viewport





Viewport Meta Tag

The viewport meta tag allows us to control:

- size of the layout viewport
- scaling

```
<meta name="viewport" content="width=device-width, user-scalable=0, initial-scale=1.0, maximum-scale=1.0;" />
```

Viewport Meta Tag

Media Queries Syntax

- Embedded in a stylesheet

```
@media [ not | only ] type [ and ] ( expression) { rules }
```

- Link external stylesheet

```
<link href="file.css" rel="stylesheet" media="[not|only] type [and] ( expression)">
```

- Import external stylesheet

```
@media url('file.css') [not|only] type [and] ( expression );
```

type : all | screen | projection | print ...

Viewport Width and Height

- **Width** : width of the browser viewport including the scroll bars
- **Height** : height of the browser viewport including the scroll bars

```
@media type and {width: value} { rules }
@media type and {min-width: value} { rules }
@media type and {max-width: value} { rules }
```

Viewport Media Query Example cssMediaQueriesViewport

```
.container {
  width: 500px;
  }
.container div {
  float: left;
  margin: 0 15px 0 0;
  width: 235px;
}
@media all and (min-width: 500px) {
  h1 {
    background: white url('background.jpg') no-repeat;
    color: black;
    height: 250px;
    padding: 20px;
    font-size: 36px;
    margin: 0px;
  }
}
```

Device Width and Height

- **Width** : width of the device screen (shorter side on iOS)
- **Height** : height of the device screen (longer side on iOS)

```
@media media and {device-width: value}
  { rules }
@media media and {min-device-width: value}
  { rules }
@media media and {max-device-width: value}
  { rules }
```

Device Media Query Example cssMediaQueriesDevice

```
.container {
  width: 500px;
  }
.container div {
  float: left;
  margin: 0 15px 0 0;
  width: 235px;
}
@media all and (max-device-width: 320px) {
  .container {
    width: auto;
  }
  .container div {
    float: none;
    margin: 0;
    width: auto;
  }
}
```

Device Orientation

- **Orientation**
  - `landscape` (viewport width > viewport height)
  - `portrait` (viewport height >= viewport width)

```
@media media and {orientation: value} { rules }
```

Orientation Media Query Example cssMediaQueriesOrientation

```css
li {
  float: left;
  border: thin solid black;
  list-style-type: none;
  padding: 10px 20px;
  text-align: center;
  max-width: 100px;
}
@media all and (orientation: portrait) {
  li { float: none; }
}
```

Device Pixel Ratio

Devices with high pixel density (> 300dpi)

Provide high-def web images

device-pixel-ratio : device pixels per CSS pixel.

```
@media media and {device-pixel-ratio: value}
  { rules }
@media media and {min-device-pixel-ratio: value}
  { rules }
@media media and {max-device-pixel-ratio: value}
  { rules }
```

```
<link rel="stylesheet" media="screen and min-device-pixel-ratio: 2" href="highres.css">
```

Avoid hiding elements if possible

```
.container {
  background: white url('background.jpg') no-repeat;
}
@media all and {max-device-width: 400px} {
  .container {
    display: none;
  }
```

Assets are downloaded even if hidden.

This approach is consuming bandwidth and cache.

Best practices

Mobile up

- Build the mobile experience first
- Use media queries to enrich and adjust the layout for larger screens
- With this approach browser that don't support media queries will get the basic stylesheet

```
<link href="basic.css" rel="stylesheet" media="screen">
<link href="desktop.css" rel="stylesheet" media="screen and (min-device-width: 480px)">
```

Best practices

Mobile up

- Build the mobile experience first
- Use media queries to enrich and adjust the layout for larger screens
- With this approach browser that don't support media queries will get the basic stylesheet

```
<link href="basic.css" rel="stylesheet" media="screen">
<link href="desktop.css" rel="stylesheet" media="screen and (min-device-width: 480px)">
```

Same concept applies for high definition graphics.

```
E { background-image: url('background-lowres.png'); }

@media all and (min-device-pixel-ratio: 1.5) {
  background-image: url('background-highres.png');
  background-size: 100% 100%; /* Ensure the images are not displayed bigger than their element */
}
```

Breakpoints

```
/* Smartphones (portrait and landscape) */
@media screen and (min-device-width : 320px) and (max-device-width : 480px) { ...  }

/* Smartphones (landscape) */
@media screen and (min-width : 321px) { ... }

/* Smartphones (portrait)  */
@media screen and (max-width : 320px) { ... }

/* iPads (portrait and landscape) */
@media screen and (min-device-width : 768px) and (max-device-width : 1024px) { ... }

/* iPads (landscape)  */
@media screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) { ...
}

/* iPads (portrait)  */
@media screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) { ...
}

/* Desktops and laptops */
@media screen and (min-width : 1224px) { ... }

/* Large screens  */
@media screen and (min-width : 1824px) { ... }

/* iPhone 4 */
@media screen and (-webkit-min-device-pixel-ratio : 1.5), screen and (min-device-pixel-ratio : 1.5) { ... }
```

Hiding/showing images responsiveMatchMedia

```
%ul.albums(data-role="listview")
  %li
    %img(src="cover1.jpg")
    Nirvana
  %li
    %img(src="cover2.jpg")
    The Beatles
  %li
    %img(src="cover3.jpg")
    Madonna
  %li
    %img(src="cover4.jpg")
    Lady Gaga
```

```
%ul.albums(data-role="listview")
  %li(data-img = "cover1.jpg")
    Nirvana
  %li(data-img = "cover2.jpg")
    The Beatles
  %li(data-img = "cover3.jpg")
    Madonna
%li(data-img = "cover4.jpg")
```

Large images can impact the loading performance

matchMedia W3C

matchMedia allows us to execute a CSS media query from Javascript and receive information if the query is a match or not

```
$(function() {
  if(window.matchMedia(" (min-width: 50em) ").matches) {
    $("[data-img]").each(function(i,val) {
      var img = new Image();
      img.src = $(val).data('img');
      $(val).prepend(img);
    })
    $('.albums').listview('refresh');
  }
});
```

Responsive Images W3C Proposal

A method for specifying multiple sources for an image through the CSS media queries

Not yet supported in browsers.

```
<picture width="500" height="500">
  <source media="(min-width: 45em)"
                          srcset="large-1.jpg 1x, large-2.jpg 2x">
  <source media="(min-width: 18em)"
                          srcset="med-1.jpg 1x, med-2.jpg 2x">
  <source srcset="small-1.jpg 1x, small-2.jpg 2x">
  <img src="fallback.jpg" >
</picture>
```

Limitations of responsive design

Responsive design excels on the visual layout but it doesn't do well:

- Content adaptation
- Performance optimizations (e.g. images)
- Optimizing heavy JS code
- Targeting low-end devices (the majority)
- Reducing latency
- Reducing loading time

Optimization

- Look at the user agent string, interpret it via a device detection repository (e.g. WURFL) and decide what to serve to that particular client (server-side)

Optimization

- Look at the user agent string, interpret it via a device detection repository (e.g. WURFL) and decide what to serve to that particular client (server-side)
- Test on the client-side what features are available (e.g. with Modernizr) and
  - take some actions on the client-side based on that information
  - inform the server that will take care of serving the right content

User Agent String

User Agent String

```
GET / HTTP/1.1
Host: example.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko)
Chrome/17.0.963.79 Safari/535.11
```

User Agent String

```
GET / HTTP/1.1
Host: example.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko)
Chrome/17.0.963.79 Safari/535.11
```

```
User-Agent: Mozilla/5.0 (iPhone Simulator; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46
(KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
```

User Agent String

```
GET / HTTP/1.1
Host: example.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko)
Chrome/17.0.963.79 Safari/535.11
```

```
User-Agent: Mozilla/5.0 (iPhone Simulator; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46
(KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
```

```
User-Agent: Mozilla/5.0 (MeeGo; NokiaN9) AppleWebKit/534.13 (KHTML, like Gecko) NokiaBrowser/8.5.0
Mobile Safari/534.13
```

User Agent String

```
GET / HTTP/1.1
Host: example.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko)
Chrome/17.0.963.79 Safari/535.11
```

```
User-Agent: Mozilla/5.0 (iPhone Simulator; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46
(KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
```

```
User-Agent: Mozilla/5.0 (MeeGo; NokiaN9) AppleWebKit/534.13 (KHTML, like Gecko) NokiaBrowser/8.5.0
Mobile Safari/534.13
```

```
User-Agent: Opera/9.80 (Linux armv7l; U; en) Presto/2.9.201 Version/11.50
```

User Agent String

```
GET / HTTP/1.1
Host: example.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.11 (KHTML, like Gecko)
Chrome/17.0.963.79 Safari/535.11
```

```
User-Agent: Mozilla/5.0 (iPhone Simulator; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46
(KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
```

```
User-Agent: Mozilla/5.0 (MeeGo; NokiaN9) AppleWebKit/534.13 (KHTML, like Gecko) NokiaBrowser/8.5.0
Mobile Safari/534.13
```

```
User-Agent: Opera/9.80 (Linux armv7l; U; en) Presto/2.9.201 Version/11.50
```

```
Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1
Mobile/9A334 Safari/7534.48.3
```

The headache with the iPad mini

Everything on the iPad mini looks 20% smaller and there is no client or server side detection possible.

iPad2

- Display: 768 x 1024, **9.7"** (**132ppi**)
- Viewport: 768 x 1024
- Device pixel ratio: 1
- ```
  Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X)
  AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1
  Mobile/9A334 Safari/7534.48.3
  ```

iPad mini

- Display: 768 x 1024, **7.9"** (**162ppi**)
- Viewport: 768 x 1024
- Device pixel ratio: 1
- ```
  Mozilla/5.0 (iPad; CPU OS 6_0 like Mac OS X)
  AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0
  Mobile/10A406 Safari/8536.25
  ```

Position: fixed cssPositionFixed

Block the position of an element

Used for creating headers and footers

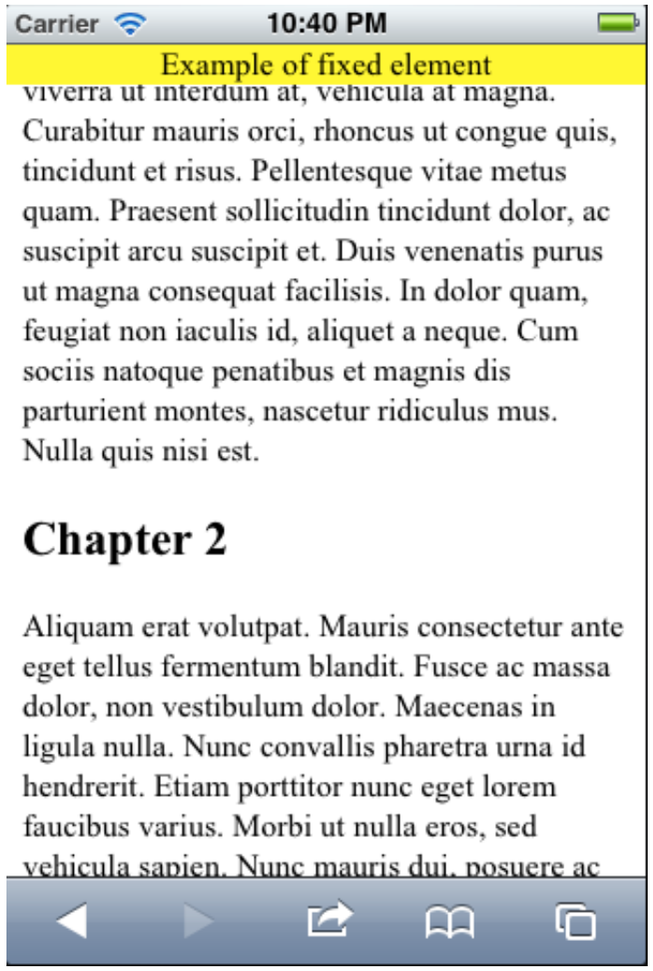
Supported on iOS 5 and Android >2.2

```
#fixed {
  top: 0;
  left: 0;
  width: 100%;
  position: fixed;
  background-color: yellow;
  text-align: center;
}
```

```
<h1>Lorem Ipsum</h1>

<div id="fixed">Example of fixed element</div>

...
```



Overflow scroll cssOverflowScroll

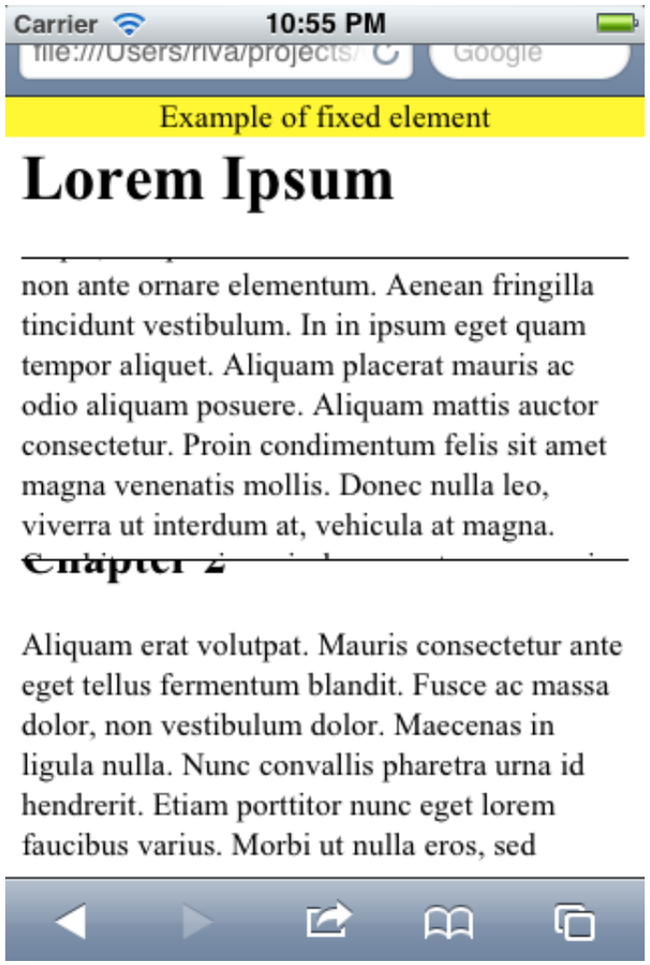
`Overflow: scroll` defines how a box should be displayed with overflowing content

`-webkit-overflow-scrolling: touch` enables native scrolling of an elemnent

Supported on iOS 5 (Check also iScroll)

```
.scroll {
  height: 150px;
  overflow-y: scroll;
  -webkit-overflow-scrolling: touch;
  border-top: 1px solid black;
}
```

```
<div class='scroll'>
  <h2>Chapter 1</h2>
  <p>Lorem ipsum dolor...
</div>
...
```



Interaction Design

Interaction Design

> Interaction design is about shaping digital things for people's use

- It's about digital products or services
- It's about satisfying people's needs
- It's about shaping a unique interaction technique
- it's about providing an enjoyable user experience

Interaction Modes - Mouse

Interaction Modes - Keyboard

Interaction Modes - Touch

Mouse events

- mousedown
- mousemove
- mouseup
- mouseover
- mouseout

Keyboard events

- keydown
- keypress
- keyup

Touch events

- touchstart
- touchmove
- touchend

Touch Events W3C

Touch events:

- **touchstart** : a finger is placed on a DOM element
- **touchmove** : a finger is moved around over a DOM element
- **touchend** : a finger is removed from a DOM element

Touches Lists

Each touch event contains three lists of touches data:

- **touches** : a list of fingers that are currently on the screen
- **targetTouches** : a list of fingers on the current DOM element
- **changedTouches** : a list of fingers that are involved in the current event

```
$(document).bind('touchmove',function(e){
    e.preventDefault();
    var touch = e.originalEvent.changedTouches[0];
    console.log(touch.pageX);
}
```

Touch Interface

- **identifier** : a unique identifier of the finger in the touch session
- **clientX, clientY** : coordinates of point relative to the viewport, excluding scroll offset
- **pageX, pageY** : coordinates of point relative to the viewport, including scroll offset
- **screenX, screenY** : coordinates of point relative to the screen
- **target** : the DOM element that was the target of the touch (even if the finger has moved out from the element)

Example of Multi-touch App canvasTouch

```
//At first touch initialize a new entry in the fingers list
$("#canvas").bind('touchstart', function(e) {
  $.each(e.originalEvent.changedTouches, function(index, v) {
    fingers[v.identifier] = {oldX: v.pageX, oldY: v.pageY, x: v.pageX, y: v.pageY};
    fingers[v.identifier].color = colors[Math.floor(Math.random() * colors.length)];
  });
});
//Track the location of the fingers
$("#canvas").bind('touchmove', function(e) {
  e.preventDefault();
  $.each(e.originalEvent.changedTouches, function(index, value) {
    fingers[value.identifier].x = value.pageX;
    fingers[value.identifier].y = value.pageY;
  });
});
//Remove the finger from the list
$("#canvas").bind('touchend', function(e) {
  $.each(e.originalEvent.changedTouches, function(index, value) {
    delete fingers[value.identifier];
  });
});
```

Example of Multi-touch App canvasTouch

```
//Draw on the canvas every 15ms (about 60fps).
var timer = setInterval(function() {
  $.each(fingers, function(index, value) {
    if (value.oldX != value.x || value.oldY != value.y ) {
      ctx.beginPath();
      ctx.moveTo(value.oldX, value.oldY);
      ctx.lineWidth = 1;
      ctx.strokeStyle = value.color;
      ctx.lineTo(value.x,value.y);
      ctx.closePath();
      ctx.stroke();
      value.oldX = value.x;
      value.oldY = value.y;
    }
  });
}, 15);
```

Interaction Modes

3 interaction modes

3 set of events

What's the best way to handle them ?

- Sometimes we can group under the same approach
- Sometimes we must handle then independently

Events sequence touchEvents

What happens when you touch the screen of a touch device ?

Click

- touchstart
- (mouseout)
- mouseover
- mousemove (once)
- mousedown
- mouseup
- click
- touchend

Double tap

- touchstart
- touchend
- touchstart
- touchend

Move finger

- touchstart
- touchmove
- touchmove
- ...
- touchmove
- touchend

Support both mouse and touch events touchDrag

```
var square = $('.square');

square.on('touchstart', function(e){
  setOffset(e);
  square.on('touchmove', drag);
});

square.on('touchend', function(e) {
  square.off('touchmove');
});

square.on('mousedown', function(e) {
  setOffset(e);
  square.on('mousemove', drag);
});

square.on('mouseup', function(e) {
  square.off('mousemove');
});
```

Disable mouse events when touch is detected touchDrag

```
square.on('touchstart', function(e){
  setOffset(e);
  square.on('touchmove', drag);
  square.off('mousedown');
});
```

Event equivalencies

Mouse	Keyboard	Touch
mousedown	keydown	touchstart
mousemove	keydown/press	touchmove
mouseup	keyup	touchend

Tricky Parts

Multi touch is only available for touch devices

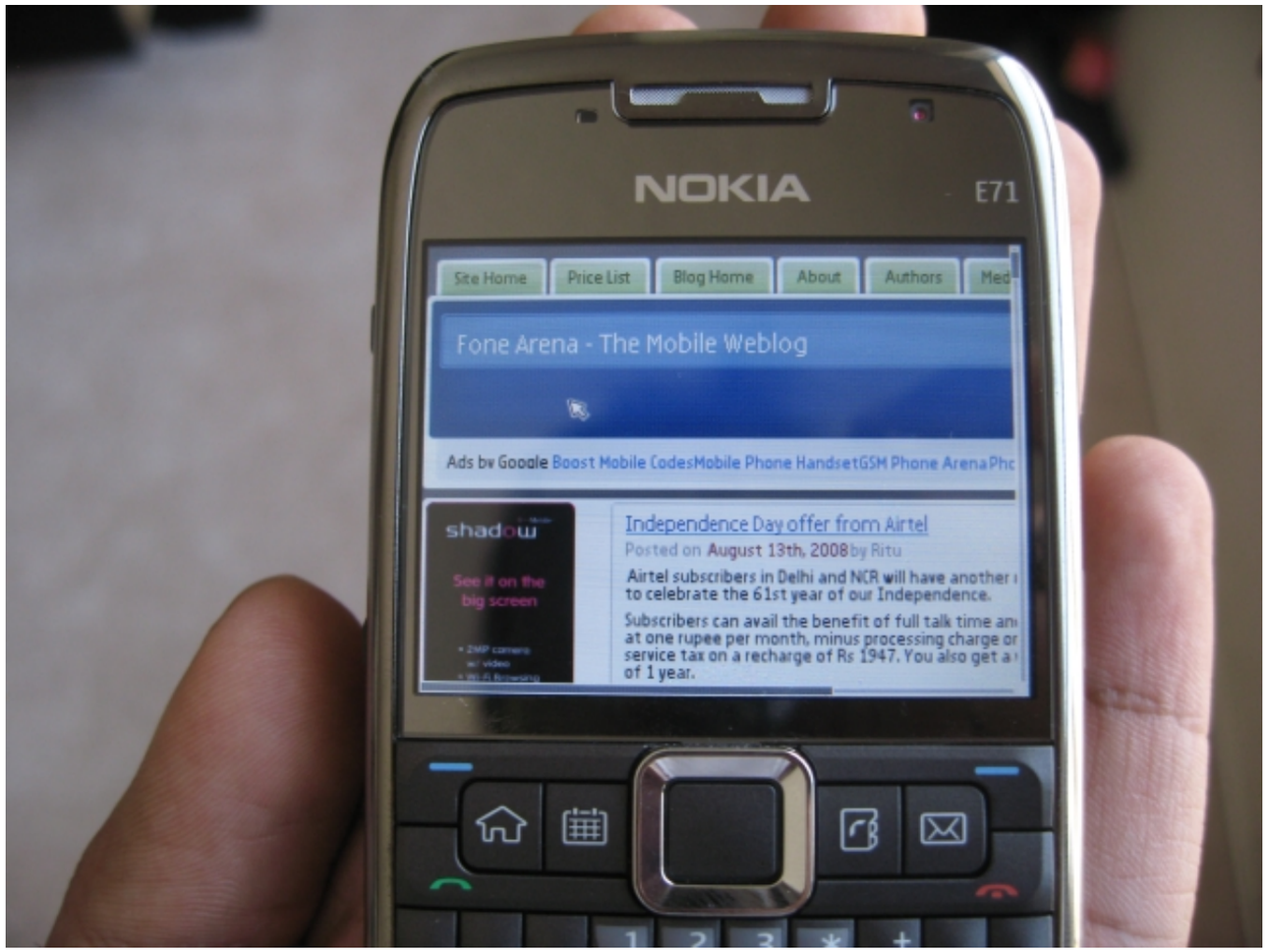
Some devices may fire multiple types of events simultaneously

Browsers on touch devices also generate click events because websites depend on them

No hover events for touch devices

Multiple interaction modes must be well designed and tested

Multiple interaction modes



Multiple interaction modes

Click event

Click is not just a mouse event

Click is synthesized by the browser

It means "Activate" the element

Works well in most of the situations

It doesn't give the best UX on touch devices

=> About 300ms delay between the touchend and the click event

Some hits for touch events

Disable zooming

```
%meta(name="viewport" content="width=device-width, user-scalable=0, initial-scale=1.0, maximum-scale=1.0;")
```

Prevent scrolling

```
$("#canvas").bind('touchmove', function(e) {
  e.preventDefault();
  ...
});
```

Asynchronously handle touch information (timer or animation frame)

jQuery Mobile provide virtual click, tap and swipe events

Cascading Style Sheets Level 3 W3C

CSS is the default styling language for every markup-based document

The version of CSS in current use is CSS2.1

CSS3 consists of several modules that are worked out and implemented independently

CSS3 is under active development

What's available in the browsers ? Can I Use

The horror of the CSS3 prefixes

Browsers specific prefixes for the CSS properties are used for implementing experimental properties

Prefixes allow browsers to modify the "experimental" properties (read more)

```
E {
  -moz-transform: function(value); /* Firefox */
  -ms-transform: function(value);  /* IE */
  -o-transform: function(value);  /* Opera */
  -webkit-transform: function(value); /*Webkit*/
  transform: function(value);
}
```

One workaround: mixins with SASS

CSS Selectors

A CSS selector consist of

- A **pattern** that is matched against all elements in the document tree
- A **rule** that is applied to the elements that match

Two main categories of selectors:

- **DOM selectors** : class, id, type, attribute selectors
- **Pseudo-selectors** : first letter of a paragraph

Versions

- CSS1 introduced the first 5-6 selectors
- CSS2 introduced 12 more selectors
- CSS3 is adding a dozen more selectors

CSS2 DOM Selectors W3C

Selector	Pattern	Description
Universal	*	Match any element
Type	E	Match any E element
Class	.title	Match any element whose class attribute contains title
ID	#header	Match any element with an id equal to header
Descendant	E F	Match any F that is a descendant of E
Child	E > F	Match any F that is a child of E
Adjacent	E + F	Match any F that has is immediately preceded by a sibling E

CSS2 DOM Selectors - Example

```css
.title { font-weight: bold; }

#header { position: fixed; }

#header .title { color: #f00; }

p.important { color: #f00; }
```

CSS2 DOM Selectors W3C

Selector	Pattern	Description
Attribute	E[attr]	Match any E that has attribute attr
Exact Attribute	E[attr='val']	Match any E that has attribute attr to be equal to val
Partial Attribute	E[attr~='val']	Match any E where val is one of the value in the list of space-separated values of attr
Lanuage Attribute	E[attr|='val']	Match any E where attr is a hyped-separated list of values that beging with val

CSS2 DOM Selectors - Example

```html
<ul>
  <li><a href="" lang="en-GB" data-options="internal">Internal</a></li>
  <li><a href="" lang="es-ES" data-options="internal hybrid">Hybrid</a></li>
  <li><a href="" lang="es-MX" data-options="external">External</a></li>
</ul>
```

```css
/* Color red all anchors with the data-options attribute */
a[data-options] { color: #f00; }

/* Color red all anchors where data-options equals internal */
a[data-options='internal'] { color: #f00; }

/* Color red all anchros where data-options contains internal */
a[data-options~='internal'] { color: #f00; }

/* Color red all anchors where lang stars with es */
a[lang|='es'] { color: #f00; }
```

CSS3 DOM Selectors W3C

Selector	Pattern	Description
Beginning	E[attr^='val']	Match any E whose attr attribute starts with val
Ending	E[attr$='val']	Match any E whose attr attribute ends with val
Arbitrary	E[attr*='val']	Match any E whose attr attribute contains the substring val
Multiple	E[attr^='val1'] [attr*='val2']	Match any E where attr starts with val1 and contains the substring val2

CSS3 DOM Selectors cssSelectors

```css
a[href^='mailto'] {
  background: url('...') no-repeat left center;
  padding-left: 20px;
}

a[href^='http'] {
  background: url('...') no-repeat left center;
  padding-left: 20px;
}

a[href$='.pdf'] {
  background: url('...') no-repeat left center;
  padding-left: 20px;
}

a[href$='.doc'] {
  background: url('...') no-repeat left center;
  padding-left: 20px;
}

a[href*='.rss'] {
  background: url('...') no-repeat left center;
  padding-left: 20px;
}
```

CSS3 Sibling Combinator W3C

Selector	Pattern	Description
Descendant	E F	Match any F that is a descendant of E
Child	E > F	Match any F that is a child of E
Adjacent	E + F	Match any F that is immediately preceded by a sibling E
General	**E ~ F**	**Match any F that is preceded by a sibling E regardless of whether it is immediately adjacent**

CSS3 Sibling Combinator - Example

HTML

```
%p.text2 This text is not affected by the rules
%p.text1 This text has no style
%p.text2 This text is affected by both rules
%div
  %p This text is on a different level
%p.text2 This text is affected by the second rule
```

CSS

```
p.text1 + p.text2 { font-weight: bold; }
p.text1 ~ p.text2 {font-style: italic; }
```

This text is not affected by the rules
This text has no style
This text is affected by both rules
This text is on a different level
This text is affected by the second rule

CSS2 Pseudo-selectors W3C

Pseudo-selectors match elements based on information that is not available in the document tree but comes from the state of the elements or their relative position

Selector	Pattern	Description
First Child	E:first-child	Match E where E is the first child of its parent
Link	E:link E:visited	Match anchor E that is not visited or already visited
Dynamic	E:active E:hover E:focus	Match E during certain user actions
Language	E:lang(fr)	Match all E that are in language 'fr'

CSS3 Pseudo-selectors W3C

Selector	Pattern	Description
First/Last Child	E:first-child E:last-child	Match E where E is the first/last child of its parent
nth Child	E:nth-child(n) E:nth-last-child(n)	Match E that is the nth child or nth child counting from the last of its parent
Only Child	E:only-child	Match E that that has no siblings
First/Last Sibling	E:first-of-type E:last-of-type	Match E where E is the first/last child of its parent
nth Sibling	E:nth-of-type(n) E:nth-last-of-type(n)	Match E that is the nth sibling or nth sibling counting from the last
Only Sibling	E:only-of-type	Match E that that has no siblings of the same type of E
First line	E:first-line	Match the content of the first line of text of element E
First letter	E:first-line	Match the first letter of text of element E

CSS3 Pseudo-selectors cssSelectors

```css
/* Enlarge first line of paragraph text*/
p:first-line {font-size: 1.5em;}

/* Enlarge first letter of the paragraph */
p:first-letter {
  font-size:250%;
  font-weight:bold;
}

/* Enlarge first letter of the paragraph */
p:nth-child(2n+1) {
  font-style: italic;
}

/* Indent the first line of paragraph except the first one*/
p + p { text-indent: 1.5em; }
```

CSS3 2D Transformations

Transformation property:

```
E { transform: function(value); }
```

```
E {
  -moz-transform: function(value); /* Firefox */
  -ms-transform: function(value);  /* IE */
  -o-transform: function(value);  /* Opera */
  -webkit-transform: function(value); /*Webkit*/
  transform: function(value);
}
```

functions:

- rotate(angle)
- translate(translateX, translateY)
- skew(skewX, skewY)
- scale(scaleX, scaleY)

CSS3 2D Transformations

```
transform: rotate(-45deg);
```

```
transform: skew(-20deg, 10deg);
```

```
transform:  scale(2,0.5);
```

CSS3 2D Transformations

Elements retain their position in the flow but they are rendered according to the transformations

You can change the default origin of the transformation (default is the center)

```
E {
  transform: rotate(45deg);
  transform-origin: left top;
}
```

This text will will soon rotate up and down

CSS3 Transitions

Create a smooth transition between two states of an element

A transition is trigger when a new value is set for a CSS property

```
.mytext {
  background-color: black;
  transition: background-color 4s;
}
.mytext:hover {
  background-color: silver;
}
```

Move over this text and the color will change

CSS3 Transitions

```
E {
  transition-property: keyword;
  transition-duration: time;
  transition-timing-function: keyword;
  transition-delay: time;
}
```

- **transition-property** : all | none | CSS property
- **transition-duration** : time in ms | s
- **transition-timing-function** : ease | linear | ease-in | ease-out | ease-in-out | cubic-bezier(...)
- **transition-delay** : time in ms | s

CSS3 Multiple Transitions

Create a smooth transition between two states of an element

A transition is trigger when a new value is set for a CSS property

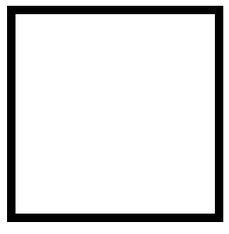
```
.mytext {
  background-color: green;
  font-size: 100%;
  color: white;
  padding-left: 10px;
  transition: background-color 4s, padding-left 4s, font-size 4s;
}
.mytext:hover {
  background-color: red;
  font-size: 200%;
  padding-left: 400px;
}
```

Animate me!

CSS3 Animations

```
@keyframes 'expand' {
  from { border-color: black; }
  50% { border-width: 10px; }
  100% {
    border-color: silver;
    width: 150px;
    background-color: green;
    transform: rotate(90deg);
  }
}
```

```
.square {
  display: block;
  border: 4px solid black;
  background-color: red;
  height: 100px;
  width: 100px;
  -webkit-animation: expand 6s ease 0 infinite
alternate;
}
```



CSS3 Animations

```
@keyframes 'name' {
  keyframe {
    property: value
  }
}
```

- **name** : name of the animation
- **keyframe** : from | to | 0% | 100% | x%

CSS3 Animations

```
E {
  animation-name: name;
  animation-duration: time;
  animation-timing-function: keyword;
  animation-delay: time;
  animation-iteration-count: count;
  animation-direction: keyword;
}
```

- **animation-name** : name of the keyframe
- **transition-duration** : time in ms | s
- **transition-timing-function** : ease | linear | ease-in | ease-out | ease-in-out | cubic-bezier(...)
- **transition-delay** : time in ms | s
- **transition-count** : 0, 1, ... | infinite
- **transition-direction** : normal | alternate