

Mobile Web Applications Development with HTML5



Lecture 5: CSS3 & Touch

Claudio Riva

Aalto University - Spring 2012

LECTURE 5: CSS3 & TOUCH

MEDIA QUERIES

POSITION: FIXED & OVERFLOW: SCROLL

TOUCH INTERACTION

CSS3 SELECTORS

CSS3 TRANSFORMATIONS

CSS3 TRANSITIONS & ANIMATIONS

The problem

The problem

```
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
```

The problem

```
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
```

The problem

```
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
```

The problem

```
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
```

The problem

```
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
```

3 categories and 3 UX factors

3 main categories of devices

- Desktop/laptops
- Tablets
- Mobile devices (from smartphones to dumbphones)

3 main factors affect UX

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

Desktop/Laptop



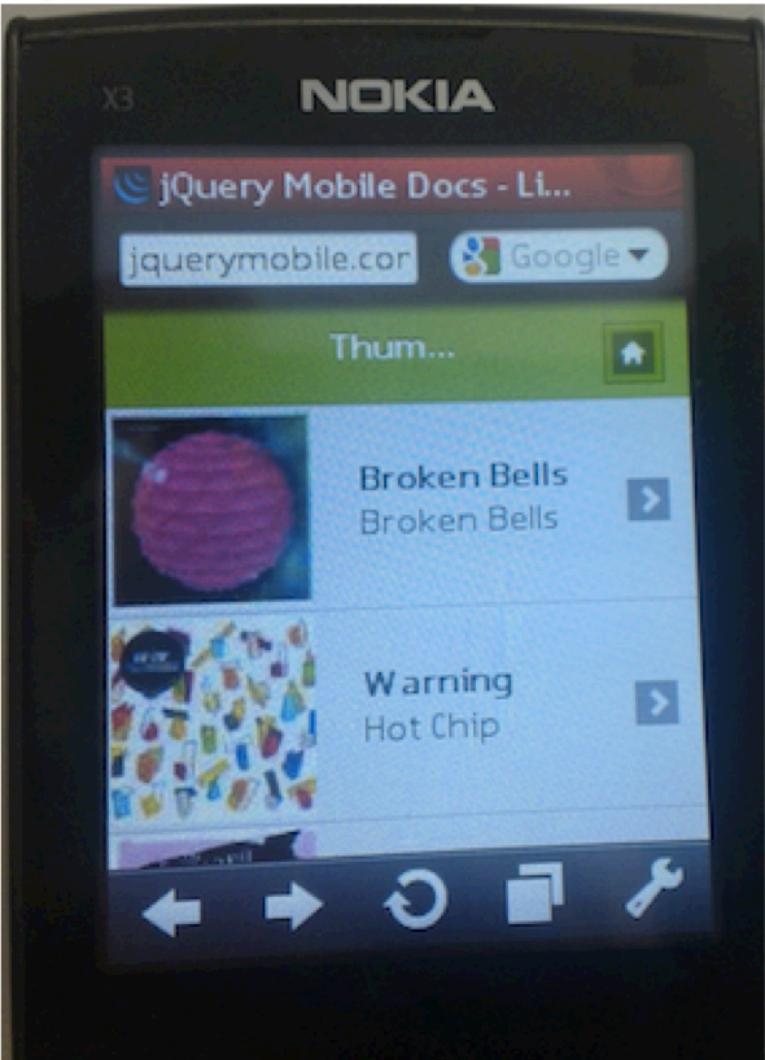
- Large screen
- High performance
- High bandwidth
- Multiple browsers with different capabilities
- Mouse and keyboard interaction (but not always)

Tablet



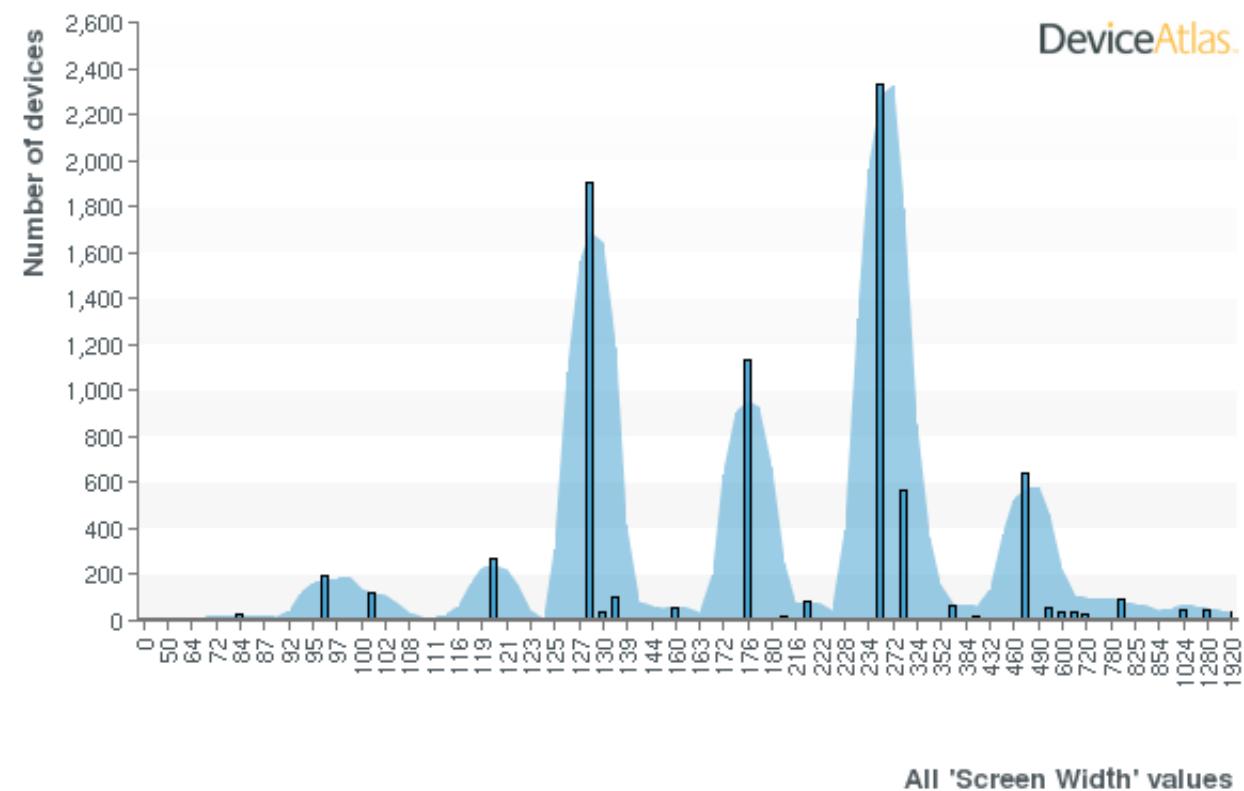
- Large screen
- Close to smartphones for user interaction
- Two orientations
- Smooth animations and graphics (often equipped with powerful GPU)
- High to medium performance
- High to medium bandwidth
- Primarily used with the default browser (but not always)
- Touch interaction (but not always)

Smartphones & Featurephones



- Small screen
- One or two orientations
- Limited animations and graphics
- Medium-low performance
- 2G/3G bandwidth
- 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

Device Screen Width



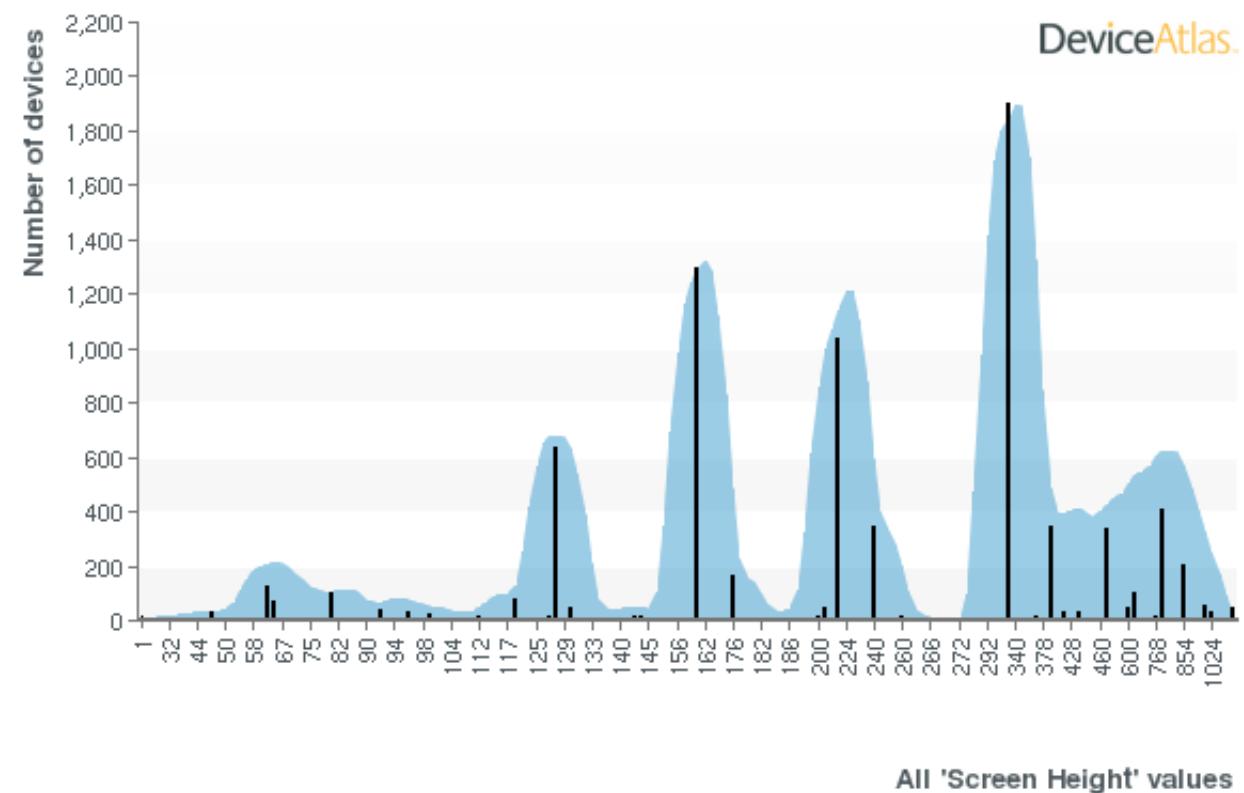
240px (28%)

128px (23%)

176px (13%)

480px (7%)

Device Screen Height



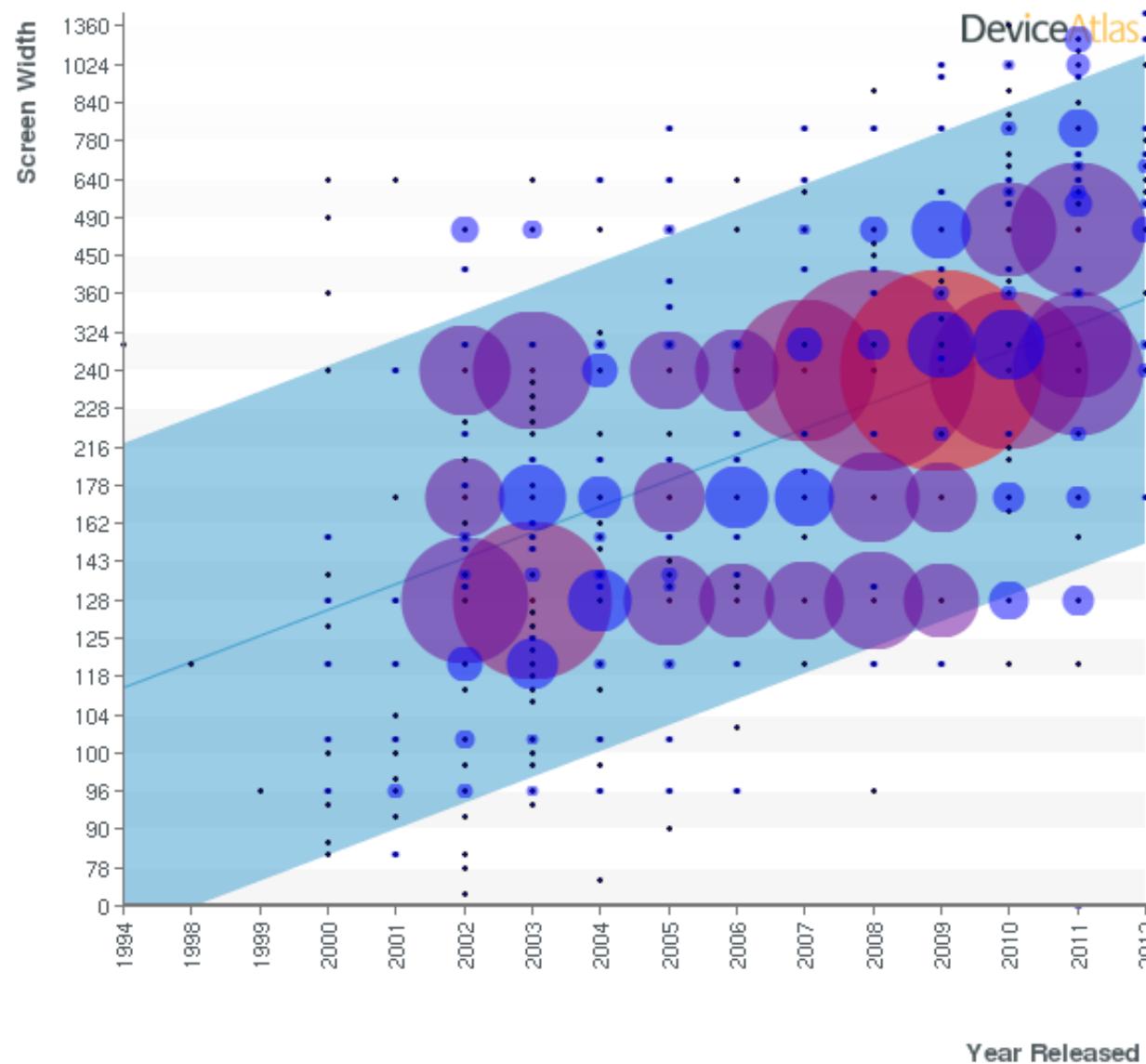
320px (23%)

160px (15.5%)

220px (12.4%)

128px (7.6%)

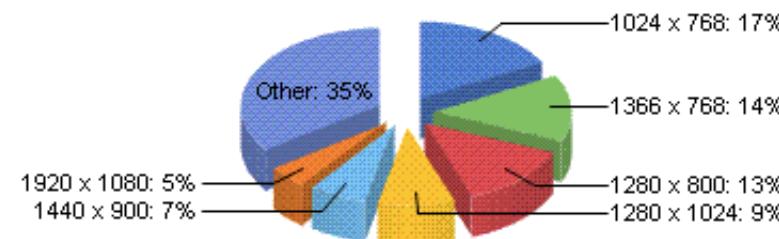
Device Screen Height over Time



2012 : 176, **240**, 320,
360, **480**, **540**, 600,
640, **720**, 768, 780,
800, 1024, 1280, 1920

2011 : 120, 128, 160,
176, 220, **240**, **320**,
360, 400, **480**, 540,
600, 640, 720, 768,
800, 840, 960, 1024,
1240, 1280

Screen Resolution Market Share



Resolution ▾	Total Market Share ▾
1024 x 768	16.22%
1366 x 768	13.49%
1280 x 800	12.72%
1280 x 1024	8.43%
1440 x 900	6.35%
1920 x 1080	4.65%
1680 x 1050	4.17%
320 x 480	3.72%
1600 x 900	3.39%
768 x 1024	3.17%
1920 x 1200	1.58%
1024 x 600	1.55%
1152 x 864	1.26%
1280 x 720	1.25%
1280 x 768	1.21%

Screen Resolution Alert on April 11

Screen Resolution Alert for Web Developers

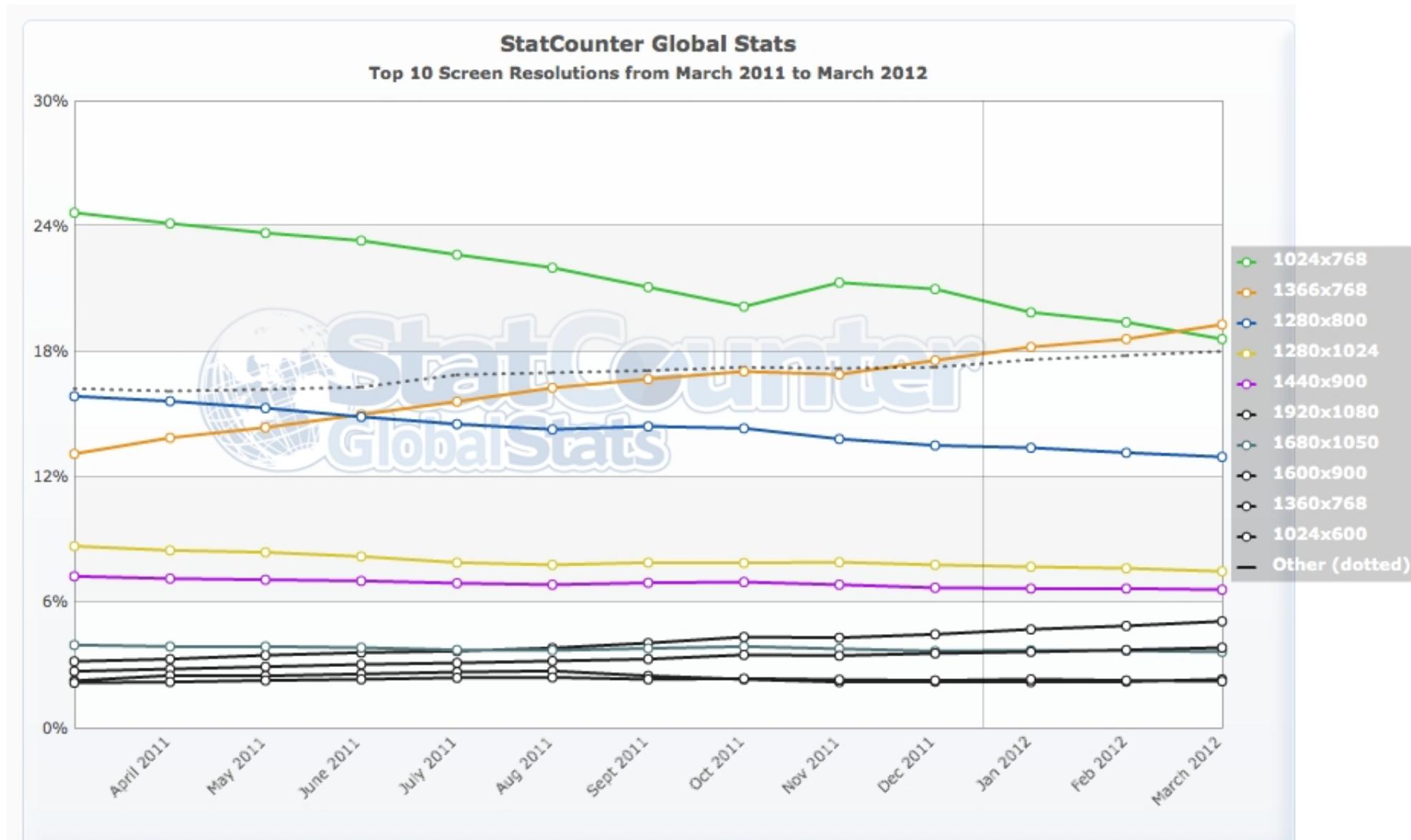
- Web analytics firm reports that 1366x768 overtakes 1024x768 for first time

Boston, USA and Dublin, Ireland, Wednesday, 11th April, 2012: A major milestone in screen resolution sizes has been passed according to independent [web analytics](#) company, StatCounter. The company's research arm, [StatCounter Global Stats](#) reports that for the first time 1366x768 has become the most popular screen resolution worldwide, having overtaken 1024x768.

"The data reflects a continuing trend of users moving to larger screen resolution sizes," commented Aodhan Cullen, CEO, StatCounter. "The screen resolution size people are using is a critical factor for developers when it comes to web design, particularly in the case of fixed width web pages."

Since StatCounter began its tracking of screen resolution in March 2009, as a free service to developers and other users, 1024x768 has been the dominant screen size globally on the web (excluding mobile*). 1024x768 has fallen from 41.8% in March 2009 to 18.6% in March 2012. Over the same period 1366x768 has grown from 0.68% to 19.28%. The third most popular size is 1280x800 at 13%.

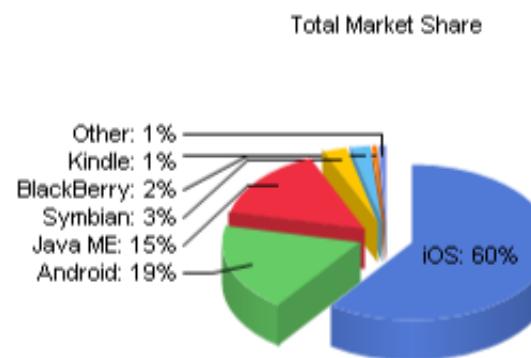
Screen Resolution Trends



Mobile OS Market Share (based on browser)

Mobile/Tablet Operating System Market Share

March, 2012

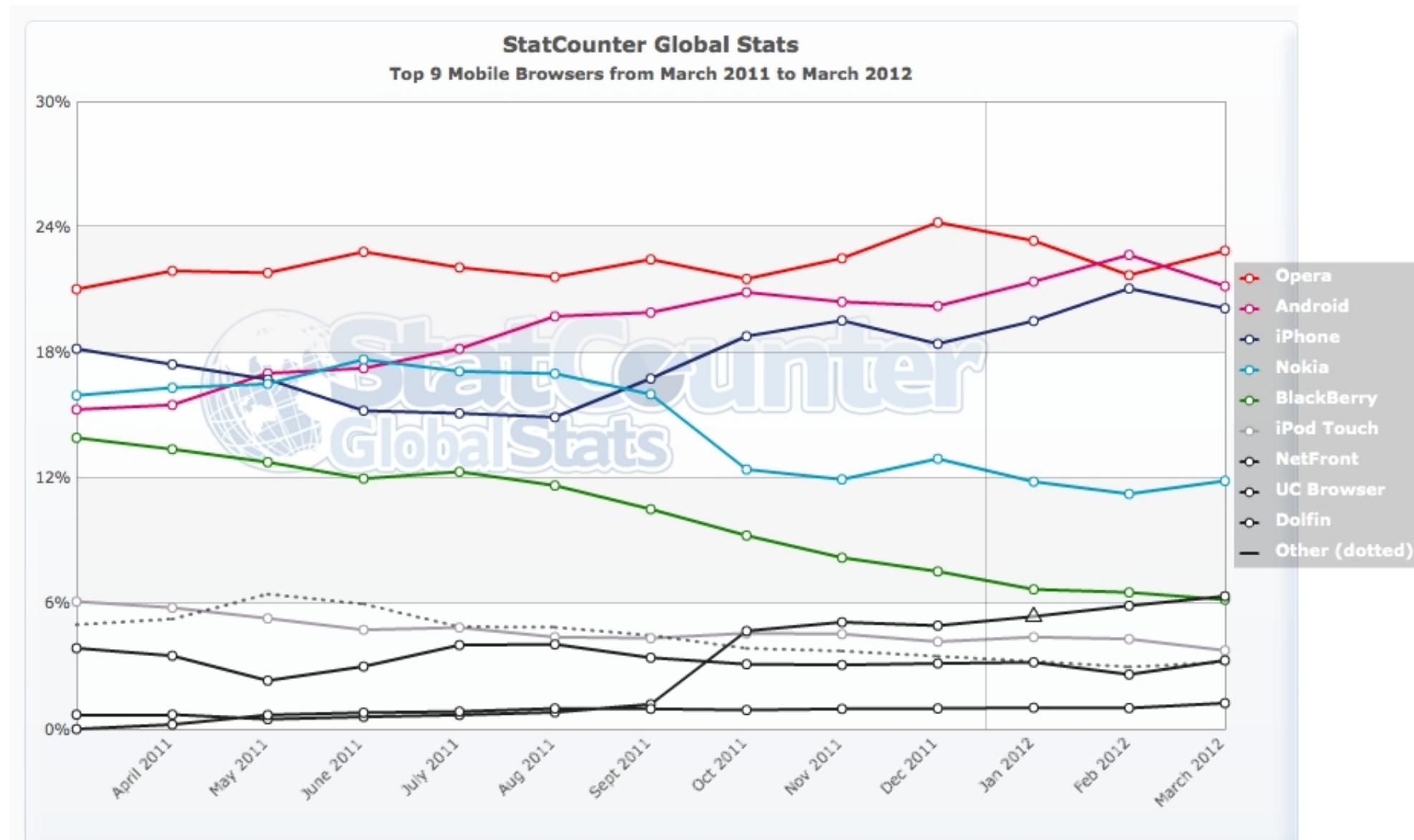


Operating System	Total Market Share
iOS	59.87%
Android	18.66%
Java ME	15.10%
Symbian	2.77%
BlackBerry	2.21%
Kindle	0.52%
Windows Phone	0.42%
Windows Mobile	0.13%
Bada	0.12%
Samsung	0.12%
BREW	0.05%
LG	0.03%
HUAWEI	0.01%
ZTE	0.00%
Palm	0.00%

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
- **1024 x 768** : iPad and Android tablets

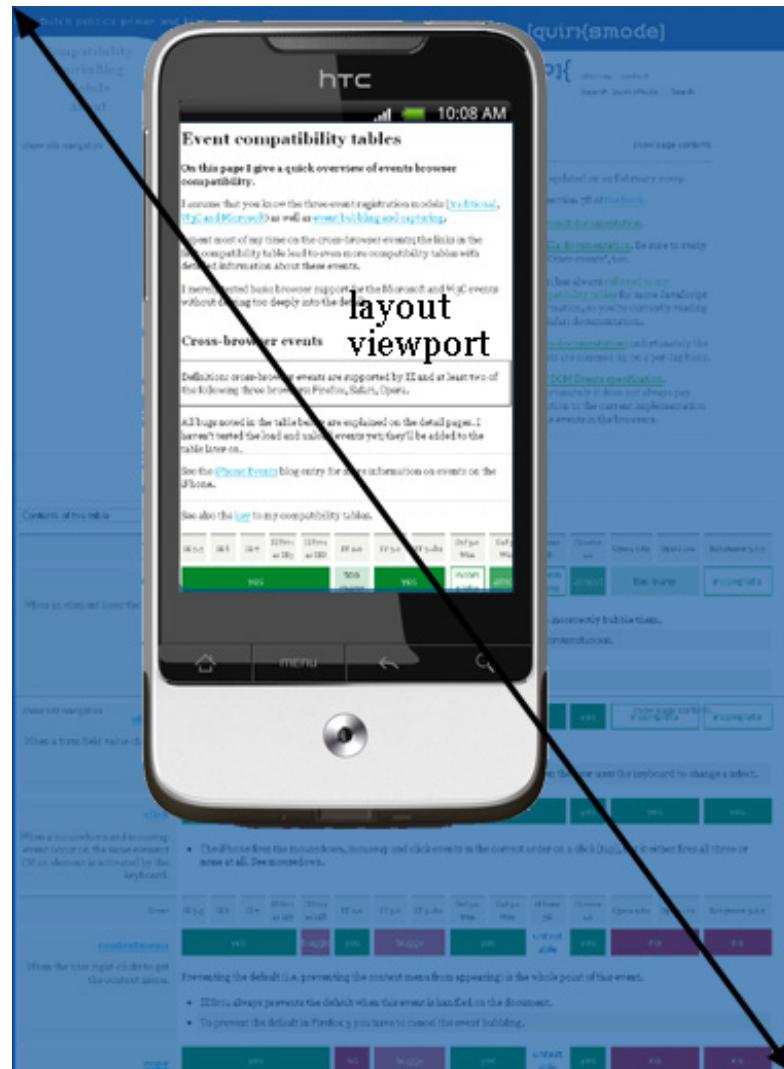
Top Mobile Browsers Trends



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



Viewport Meta Tag



```
<meta name="viewport" content="width=device-width; user-scalable=0;" />
```

⌚ 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
- to serve the mobile optimized versions (i.e. smaller images)

Responsive Web Design

- [An example](#)
- [Another example](#)
- [Other examples](#)

Media Queries Syntax

- Link external stylesheet

```
<link href="file.css" rel="stylesheet" media="logic media and ( expression)">
```

- Import external stylesheet

```
@media url('file.css') logic media and ( expression );
```

- Embedded in a stylesheet

```
@media logic media ( expression ) { rules }
```

Logic : only | not

Media : 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 media and {width: value} { rules }
@media media and {min-width: value} { rules }
@media media 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 \geq viewport width)

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

Orientation Media Query Example

cssMediaQueriesOrientation

```
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 is consuming bandwidth and cache.

Best practices

Define first the basic stylesheet for mobile users and include the one for desktop/tablets through media queries

Legacy browsers 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 is valid 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');
    /* Ensure the images are not displayed bigger than their element */
    background-size: 100% 100%;
}
```

CSS for each device layout

```
/* 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) { ... }
```

Use responsive design carefully

Layout of the page is important but building a mobile web site (app) includes also:

- Optimizing performance
- Avoiding heavy JS libraries
- Reducing latency
- Reducing loading time
- Task-oriented UI

Position: fixed cssPositionFixed

Block the position of an element

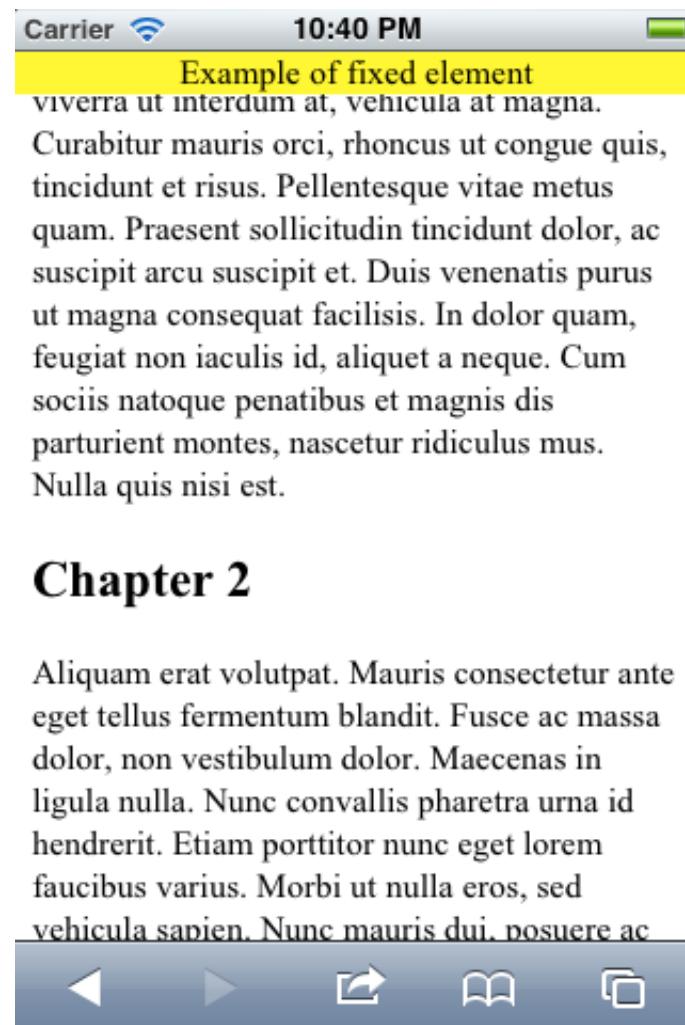
Used for creating headers and footers

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>
```

...



Chapter 2

Aliquam erat volutpat. Mauris consectetur ante eget tellus fermentum blandit. Fusce ac massa dolor, non vestibulum dolor. Maecenas in ligula nulla. Nunc convallis pharetra urna id hendrerit. Etiam porttitor nunc eget lorem faucibus varius. Morbi ut nulla eros, sed vehicula sapien. Nunc mauris dui. posuere ac

Overflow scroll cssOverflowScroll

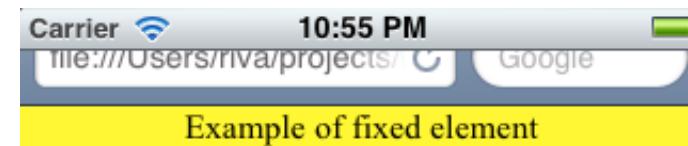
Overflow: scroll defines how a box should be displayed with overflowing content

-webkit-overflow-scrolling: touch
enables native scrolling of an element

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...</p>  
</div>  
...
```



Lorem Ipsum

non ante ornare elementum. Aenean fringilla tincidunt vestibulum. In in ipsum eget quam tempor aliquet. Aliquam placerat mauris ac odio aliquam posuere. Aliquam mattis auctor consectetur. Proin condimentum felis sit amet magna venenatis mollis. Donec nulla leo, viverra ut interdum at, vehicula at magna.

Chapter 1

Aliquam erat volutpat. Mauris consectetur ante eget tellus fermentum blandit. Fusce ac massa dolor, non vestibulum dolor. Maecenas in ligula nulla. Nunc convallis pharetra urna id hendrerit. Etiam porttitor nunc eget lorem faucibus varius. Morbi ut nulla eros, sed



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 them 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

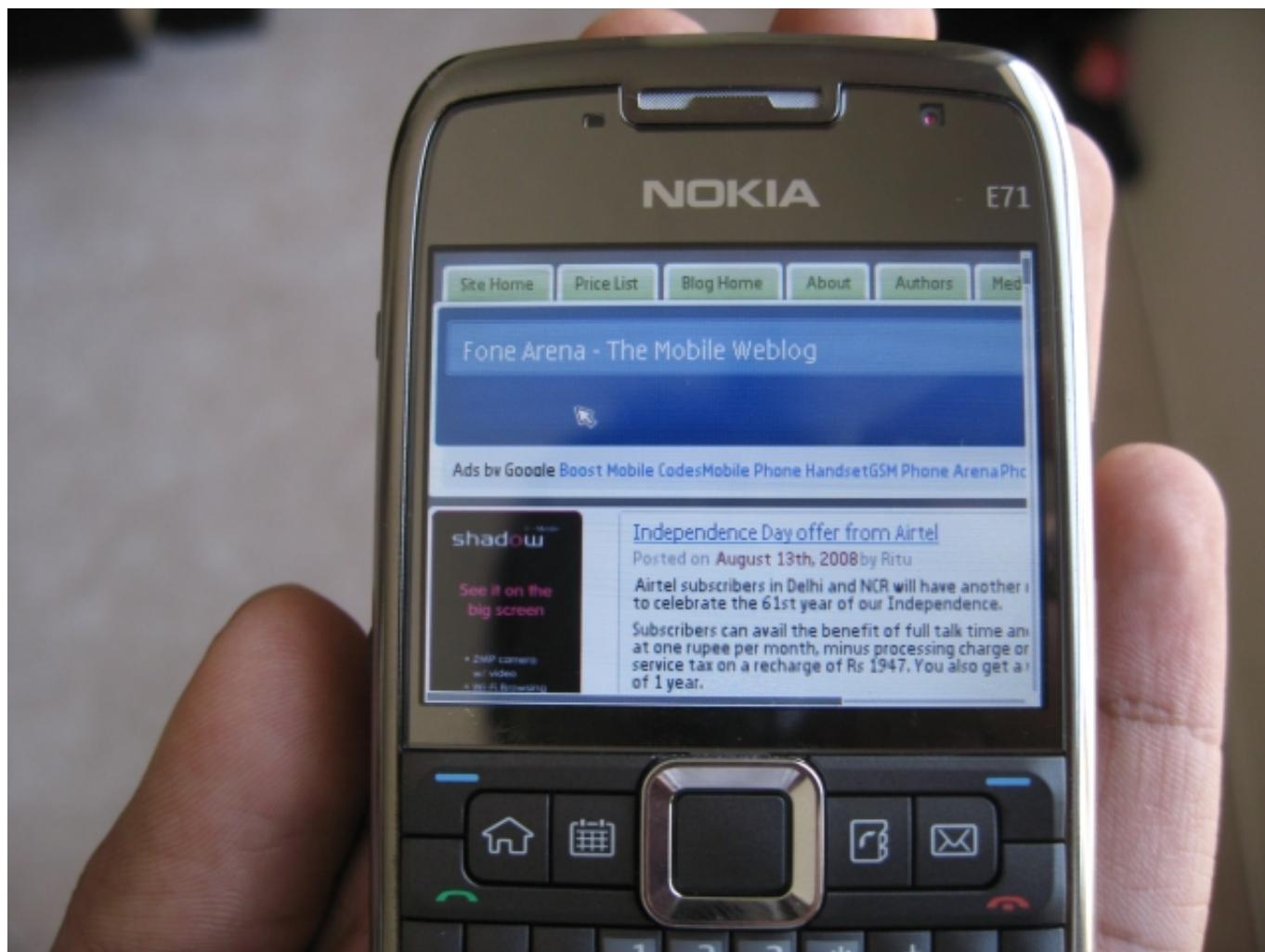
Some devices may fire multiple types of events simultaneously

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 consists 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

```
.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
Language Attribute	E[attr = 'val']	Match any E where attr is a hyphen-separated list of values that begin with val

CSS2 DOM Selectors - Example

```
<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>
```

```
/* 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 starts 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

```
a[href^='mailto'] {  
    background: url('data:image/png;base64,...') no-repeat left center;  
    padding-left: 20px;  
}  
  
a[href^='http'] {  
    background: url('data:image/png;base64,...') no-repeat left center;  
    padding-left: 20px;  
}  
  
a[href$='.pdf'] {  
    background: url('data:image/png;base64,...') no-repeat left center;  
    padding-left: 20px;  
}  
  
a[href$='.doc'] {  
    background: url('data:image/png;base64,...') no-repeat left center;  
    padding-left: 20px;  
}  
  
a[href*='.rss'] {  
    background: url('data:image/gif;base64,...') 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'

3 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 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 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

3 CSS3 Pseudo-selectors cssSelectors

```
/* 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);
```

3 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 soon *rotate up and down*

3 CSS3 Transitions

Create a smooth transition between two states of an element

A transition is triggered 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

3 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

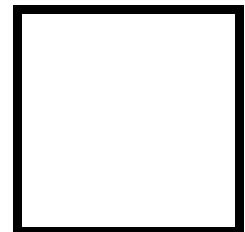
A transition is trigger when a new value is set for a CSS property

```
.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%

3 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