

Mobilize Your Website

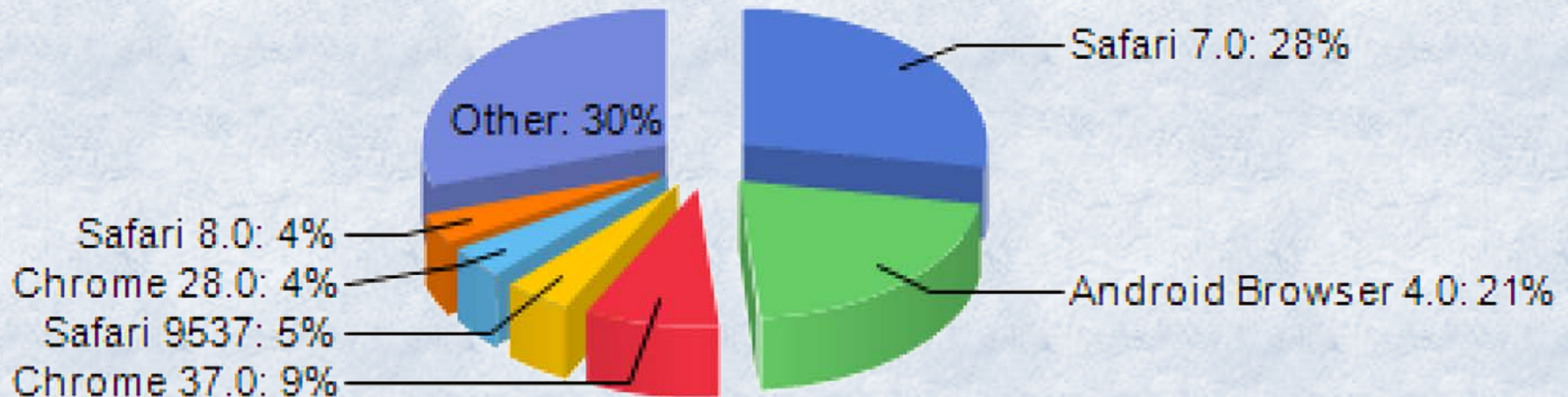
HTML5 Developer Conference
Fall 2014



shaRon sachse | @PCartiste

Mobile/Tablet Browser Version Market Share

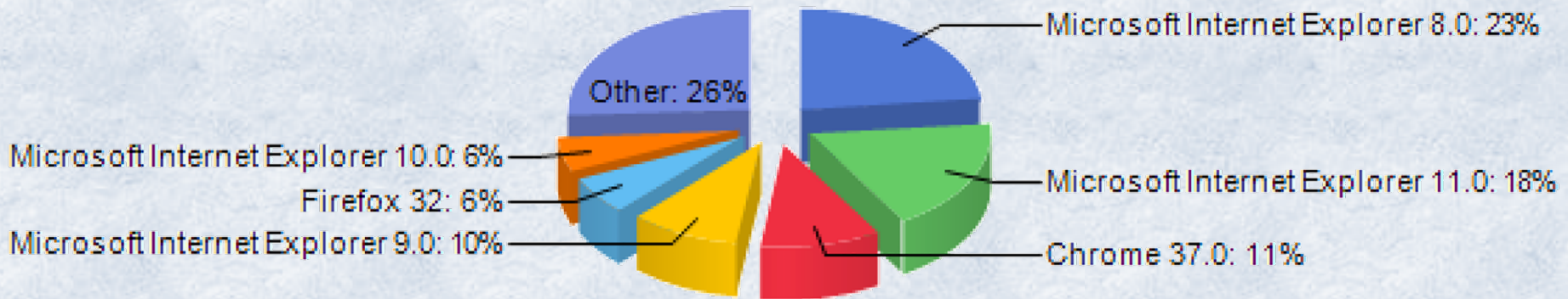
September, 2014



Browser Version	Total Market Share
Safari 7.0	28%
Android Browser 4.0	21%
Chrome 37.0	9%
Opera Mini 7.5	2.69%
IE10.0 Mobile	1.29%
Firefox 32	0.45%
BlackBerry	0.34%
Symbian	0.20%

Desktop Browser Version Market Share

September, 2014



Browser Version

IE 8

IE 11

Chrome 37

IE 9

Firefox 32

IE 10

Firefox 31

IE 6

Chrome 36.0

Safari 7.0

Opera 12.x

IE 7

Total Market Share

23%

18%

11%

10%

6%

6%

3.92%

3.23%

3.06%

2.27%

0.76%

0.68%



61% IE

Responsive Design:

Relative width layout and flexible images

Adapted with media queries

Content and design driven media queries

Adaptive Layouts:

Set or Relative width layouts

Layouts and images swapped with media queries

Device and resolution driven media queries

<http://mediaqueri.es/>

Media queries

Simple filters that can be applied to CSS styles. They make it easy to change styles based on the characteristics of the device rendering the content, including the display type, width, height, orientation and even resolution.

More diversity within devices

Not: for feeding styles based on browser

```
<link rel="stylesheet" media="only screen and (min-width:280px) and (max-width:480px)" href="css/small.css">
```

Media queries are only meant to solve the **useability** of mobile's small viewports

Not all the other things that can make mobile browsing different, such as:

- context

- bandwidth

Mobilize your site-

Typical to add on media queries for both smaller and wider styles

Can take different approach when starting from scratch

- Start with mobile

- Start with desktop

- Start with something in between for majority

```
/* Global styles will remain the same for ALL views */  
  <link rel="stylesheet" type="text/css" href="css/global.css">  
  
/* Desktop and mobile IE styles (large) */  
  <link rel="stylesheet" type="text/css" href="css/large.css">  
  
/* mobile styles (small) */  
  <link rel="stylesheet" type="text/css" media="only screen and  
  (min-width:280px) and (max-width:480px)" href="css/small.css">  
  
/* tablet styles (mid) */  
  <link rel="stylesheet" type="text/css" media="only screen and  
  (min-width:481px) and (max-width:767px)" href="css/mid.css">  
  
/* iPad styles only */  
  <link rel="stylesheet" type="text/css" media="only screen and  
  (min-width: 768px) and (max-width: 980px)" href="css/mid.css">
```


Determining **ems**

target $\dot{_}$ context = **result**

Declaring fluid margin/padding

Adjust widths so everything adds up to just **under** 100% Avoids problems due to rounding % to px

Nesting affects margin/padding values too
Use target/context formula to match outer spacing with inner spacing

Using box-sizing

Makes fixed-width margin/padding on fluid layout easy

Standard box model

box-sizing: content-box

padding & border **added** on to width/height

New box model

box-sizing: border-box

padding & border **subtracted** from width/height

Viewport meta tag

Forces mobile devices to scale viewport to actual device width

```
<meta name="viewport"  
      content="width=device-width">
```


We can pass values for:

width, height

device-width, device-height

maximum-scale, minimum-scale, initial-scale

user-scalable

```
<meta name="viewport "  
  content="maximum-scale=1.0  
  minimum-scale=1.0, initial-scale=1.0" />
```

Viewport is viewed full-scale all the time

(user-scalable=no)

Fixing zoom issues

Option 1:

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0, minimum-scale=1.0" />
```

width=device-width

browser window will match the width of the device

initial-scale=1.0

set zoom level on page load to normal (100%)

minimum-scale=1.0

tells the browser not to zoom out past 100%

Creates the expected behavior

but triggers over-zoom/crop bug
when changing from portrait to landscape in iOS

Zoom problem in iOS

The device-width on iOS devices always matches portrait width

This means design **doesn't reflow** when you switch to landscape, but instead just **zooms**

The best way to fix zoom issues

Option 2:

```
<head>
```

```
...
```

```
  <meta name="viewport" content="width=device-width,  
    initial-scale=1.0, minimum-scale=1.0" />
```

```
...
```

```
<script src="ios-orientationchange-fix.js">
```

```
<style>
```

```
html, body {-webkit-text-size-adjust: 100%;}
```

```
</style>
```

```
</head>
```

<https://github.com/scottjehl/iOS-Orientationchange-Fix>

Dealing with IE 8 and earlier [if (lt IE 9)]

add scripting to further customize the
content, markup and functionality

Conditional comments

JavaScript for non-supporting browsers (media queries)

Respond:

<https://github.com/scottjehl/Respond>

css3-mediaqueries:

<http://code.google.com/p/css3-mediaqueries-js/>

Avoid extra HTTP request by using conditional comments:

```
<!--[if (lt IE 9)&(!EMobile 7)]>  
    <script src="respond.min.js"></script>  
<![endif]-->
```


JavaScript for non-supporting browsers (HTML5 markup)

```
<!--[if lt IE 9]>  
  <script  
    src="http://html5shiv.googlecode.com/sv  
n/trunk/html5.js">  
  </script>  
<![endif]-->
```

Changing feature images

```
@media screen and (max-width: 1080px) {
```

```
...
```

```
#highlight-img { background-image: url(icon128.png); }  
}
```

```
@media screen and (max-width: 480px) {
```

```
...
```

```
#highlight-img { background-image: url(icon32.png); }  
}
```

```
.highlight img {  
  max-width: 100%;  
  height: auto;  
}
```

Swapping images on high-res displays

```
@media screen and (moz--min-device-pixel-ratio : 1.5),  
screen and (-o-min-device-pixel-ratio : 3/2),  
screen and (-webkit-min-device-pixel-ratio : 1.5),  
screen and (min-device-pixel-ratio : 1.5) {
```

```
    #highlight-img { background-image: url(icon256.png); }
```

```
}
```

Sprites

A sprite is a collection of images put into a single image

Using sprites will reduce the number of server requests and save bandwidth.

```
.social li a {  
    display: block;  
    width: 32px;  
    height: 32px;  
    background-image :url(../image/icon.png);  
}
```

```
.social li a.twitter {  
    background-position: -32px 0;  
}
```

Defines: background image, size, position (left 0px, top 0px) assumed unless stated otherwise

SVG Scalable Vector Graphic

Advantages over raster images:

- Small file sizes that compress well

- Scales to any size without losing clarity (except very tiny)

- Looks great on retina displays

- Design control like interactivity and filters

- CSS effects such as color, shadowing, transparency and animations are straightforward

- SVG images can be inlined right in the document

- They are semantic

- Provide better accessibility with the appropriate attributes

- They're vector graphics that can be infinitely scaled

```
<svg width="100" height="100">
```

```
  <circle cx="50" cy="50" r="40" stroke="#ff00ff" stroke-  
width="4" fill="yellow" />
```

Sorry, your browser does not support inline SVG

```
</svg>
```

```
<object type="image/svg+xml" data="image.svg">
```

```
Your browser does not support SVG</object>
```

HTML5 <object> tag

Note you can provide fallback text or images within the object block

academyX
expert computer training
www.academyx.com

<http://mobileX.pcartiste.com>



shaRon sachse | @Pcartiste
pcartiste@gmail.com