

Best Practices for Responsive Content

Scott DeLoach – scott@clickstart.net

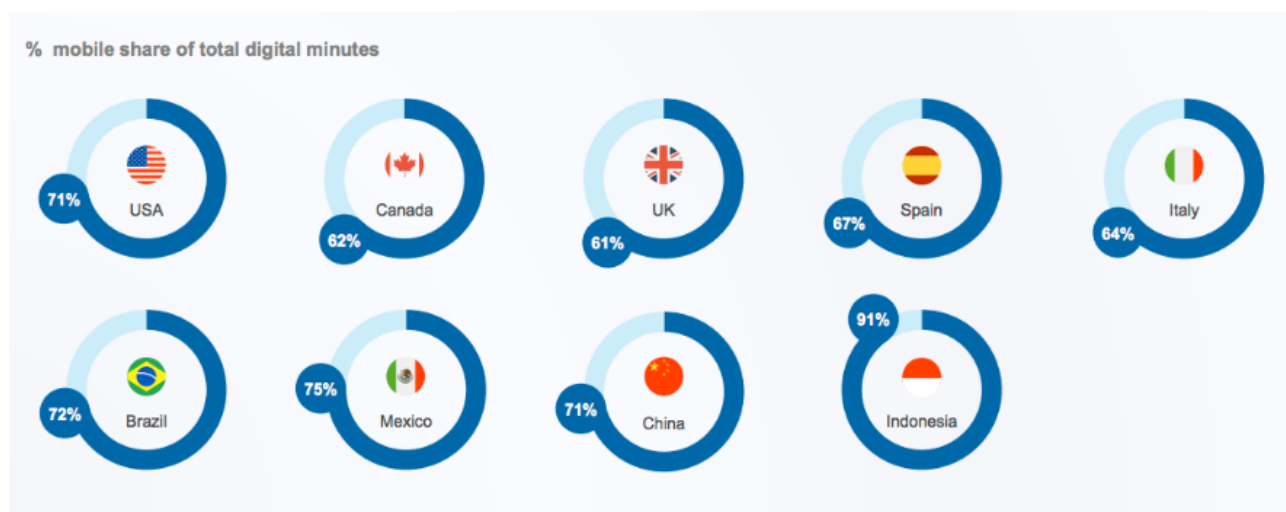
ClickStart – www.clickstart.net

In this session, I will demonstrate CSS and jQuery-based approaches for providing responsive content. Responsive content (as opposed to responsive design) focuses on content that adapts for desktops, tablets, and phones. Scott will share examples for showing/hiding, redesigning, and rewording/replacing text, images, lists, tables, and other types of content based on the user's screen width.

Overview

- Why responsive content is important
- Design challenges and the solution
- Content challenges and the solution
- Responsive content examples

Why mobile?



COMSCORE

Source: comScore MMX Multi-Platform, January 2017

© comScore, Inc. Proprietary.

source: www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/attachment/mobile-share-of-online-time-percent-2017

The problems

- Content
- Design

How going mobile changes design

Everything needs respond to screen size

- Text
- Tables

- Images
- Navigational elements

The design solution: responsive web design (RWD)

Ethan Marcotte 2010

Examples

www.bostonglobe.com

www.change.org

www.foodsense.is

How going mobile changes content

Everything needs to respond to screen size

- Terminology
- Writing style
- Information blocks
- Navigation and interactivity

The content solution: responsive web content (RWC)

- Media queries
- display property (plus visibility and height for)

RWC - the code

```
/* phone */
@media (max-width: 767px) {
  body .phone { display: block; }
  body span.phone { display: inline; }
  body img.phone { display: inline; }
  body li.phone { display: list-item; height: auto; visibility: visible; }
  body table.phone { display: table; }
  body tr.phone { display: table-row; }
}

/* tablet */
@media (min-width: 768px) and (max-width: 1279px) {
  body .tablet { display: block; }
  body span.tablet { display: inline; }
  body img.tablet { display: inline; }
  body li.tablet { display: list-item; height: auto; visibility: visible; }
  body table.tablet { display: table; }
  body tr.tablet { display: table-row; }
}

/* desktop */
@media (min-width: 1280px) {
  body .desktop { display: block; }
```

```

body span.desktop { display: inline; }
body img.desktop { display: inline; }
body li.desktop { display: list-item; height: auto; visibility: visible; }
body table.desktop { display: table; }
body tr.desktop { display: table-row; }
}

.phone { display: none; }
.tablet { display: none; }
.desktop { display: none; }
li.phone { display: block; height: 0; visibility: hidden; }
li.tablet { display: block; height: 0; visibility: hidden; }
li.desktop { display: block; height: 0; visibility: hidden; }

```

RWC words

```
<span class="desktop">Click</span><span class="phone tablet">Tap</span> Home.
```

RWC paragraphs

```

<p class="desktop">This sentence is for desktops.</p>
<p class="tablet">This sentence is for tablets.</p>
<p class="phone">This sentence is for phones.</p>

```

RWC lists

```

<ul class="desktop">
  <li>Desktop</li>
  <li>Desktop</li>
  <li>Desktop</li>
</ul>
<ul class="tablet">
  <li>Tablet</li>
  <li>Tablet</li>
  <li>Tablet</li>
</ul>
<ul class="phone">
  <li>Phone</li>
  <li>Phone</li>
  <li>Phone</li>
</ul>

```

RWC items in a list

```

<ol>
  <li>A - Desktop</li>
  <li>B - Desktop</li>
  <li class="phone">C - Phone</li>
  <li>D - Desktop</li>
</ol>

```

RWC images

```
<p></p>
```

Note: This approach is useful if you want to present difference images for phones, tablets, and/or desktops. If you want to dynamically resize images as the user resizes the browser window, you can use jQuery. For examples, see <https://goo.gl/OePX4k>

RWC tables

```
<table class="phone">
  <col />
  <col />
  <thead>
    <tr>
      <th>Phone</th>
      <th>Phone</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>phone</td>
      <td>phone</td>
    </tr>
    <tr>
      <td>phone</td>
      <td>phone</td>
    </tr>
  </tbody>
</table>
```

Note: This approach (and the rows in a table approach below), is useful if you want to show/hide a table for desktops, tablets, and/or phones. If you want to change tables to lists, set them to auto-scroll, or use other more design-based solutions, see <https://goo.gl/OePX4k>

RWC rows in a table

```
<table>
  <col />
  <col />
  <thead>
    <tr>
      <th>Heading</th>
      <th>Heading</th>
    </tr>
  </thead>
  <tbody>
    <tr class="desktop">
      <td>desktop</td>
      <td>desktop</td>
    </tr>
    <tr class="tablet">
      <td>tablet</td>
      <td>tablet</td>
    </tr>
  </tbody>
</table>
```

```
</tr>
<tr class="phone">
  <td>phone</td>
  <td>phone</td>
</tr>
</tbody>
</table>
```

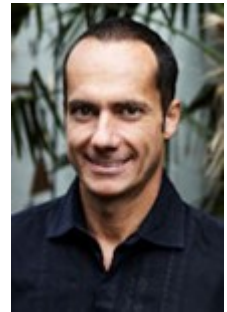
Sample project

You can download the sample project and examples at:

http://www.clickstart.net/presentations/uaeurope17/rwc_practice.zip

About the presenter

Scott DeLoach is the Founder of Click**Start**, where he provides training and consulting for MadCap Flare, Adobe Captivate, embedded user assistance, CSS, and HTML5. He has been developing browser-based help systems since 1997, and he has received four Best in Show awards for his work from STC. Scott is a certified Flare and Captivate instructor, and he is the author of *MadCap Flare 2017: The Definitive Guide*, *CSS to the Point*, *HTML5 to the Point*, and *Word 2013 to the Point*. For more information about Scott's books see <http://www.lulu.com/clickstart>.



You can reach Scott at www.clickstart.net or by email at scott@clickstart.net.