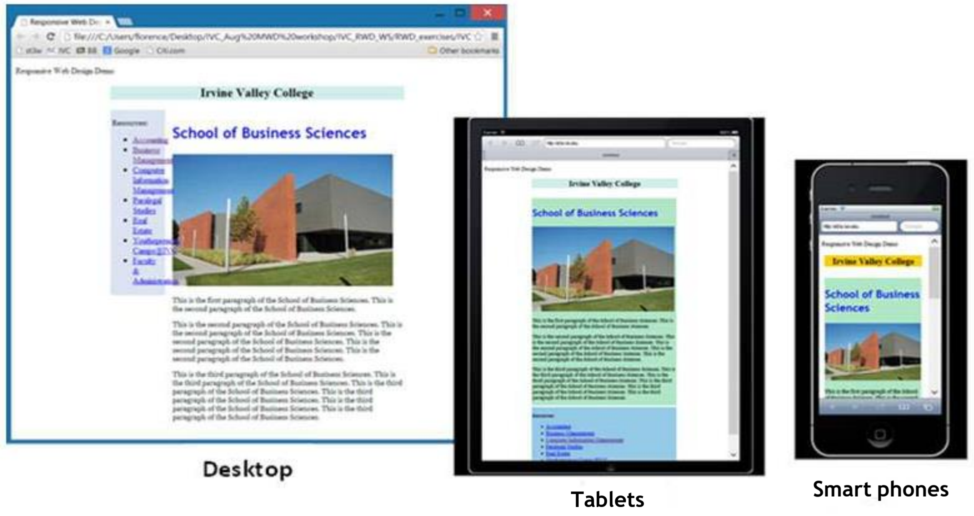
RESPONSIVE WEB DESIGN

WHAT IS RESPONSIVE WEB DESIGN (RWD)?



- A design when the layout and content adapts to the user’s devices: **screen size, platform and orientation**
- The design and development should respond to the user’s behavior
- The website that have the technology to automatically respond to the user’s preferences.

HISTORY OF THE RESPONSIVE WEB DESIGN

- The term Responsive Web Design was first coined by **Ethan Marcotte** in his article *A List Apart* in May 2010
  http://alistapart.com/article/responsive-web-design
- He defined the technique of RWD by using **fluid grids**, **flexible images**, and **media queries** to deliver different visual experiences for different screen sizes.
- Ethan expanded his RWD theory and published his book titled *Responsive Web Design*.

WHY SHOULD WE BUILD A RESPONSIVE WEB?

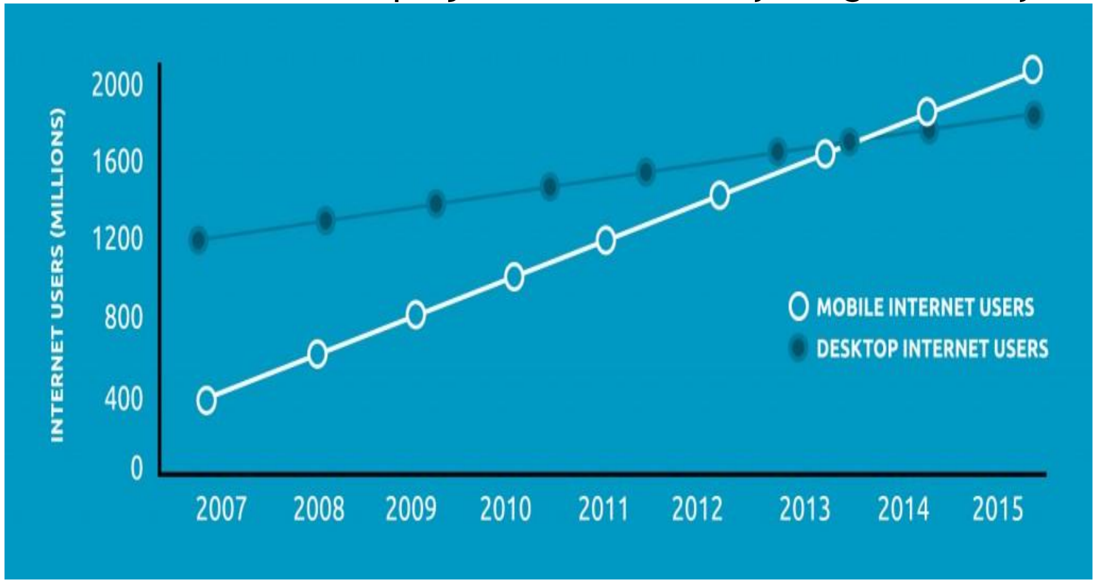
- Each year new devices are pouring into the market, responsive web design let us build one site, and modify it to adapt the new device's screen size
- Build once for all devices
- Easy to manage content editing through a single CMS (Content Management System)
- If we have a working website:
    - Not need to rebuild new websites to adapt the new devices
    - We can convert the existing working website to a responsive Web site to adapt all kind of devices

PARADIGM SHIFT TOWARDS MOBILE

- International Data Corporation predicted that by the end of 2013, tablet sales will exceed that of portable PCs.

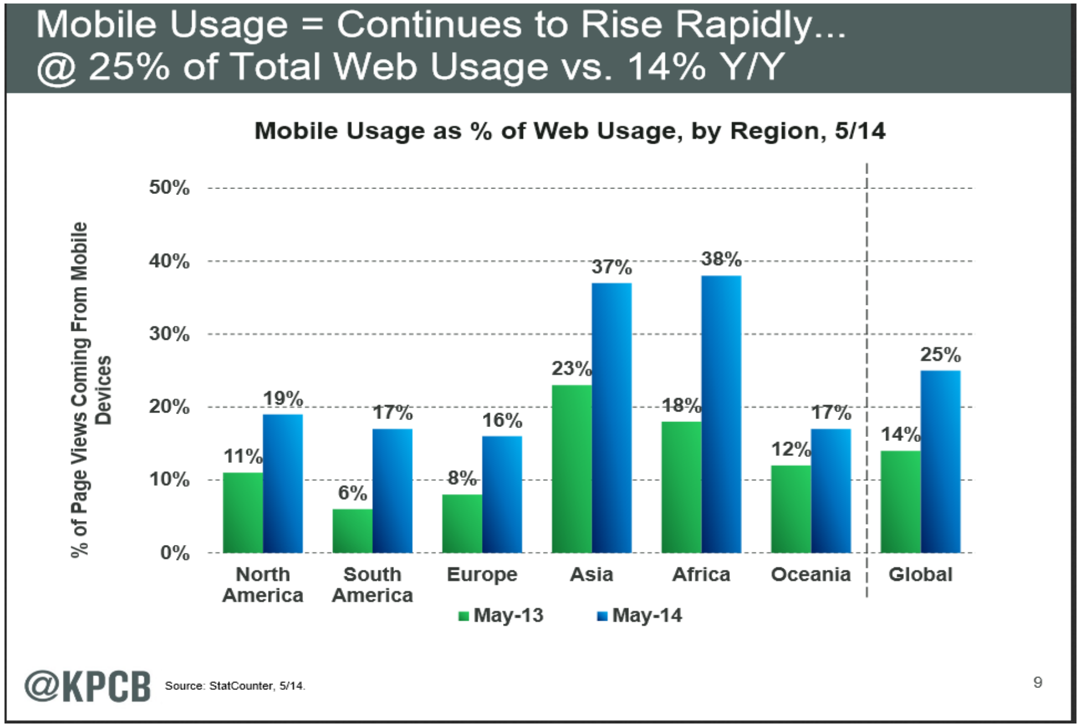
Mobiles Vs. Computers: 2007-2015

Global internet user projection research by Morgan Stanley



PARADIGM SHIFT TOWARDS MOBILE (CONTINUED)

- Mary Meeker in her 2012 *internet Trends Report* notes mobile makes up 15% of Web traffic, up from 10% a year ago
- Her recent report showed the following chart



ADVANTAGES OF RWD

- One single HTML document to be maintained
- One single CSS file to be maintained
- The site is easily accessible on any type of device.
- Better user experience.

  Users will have a similar experience using the site when they access the site from different devices.
- Responsive Web is flexible and adaptable
- Maintaining a RWD is:
  - Easier than maintaining several website for different devices.

FUNDAMENTAL TECHNIQUES FOR RWD

- There are three parts in Responsive Wed design:

1. **Flexible, grid-based layouts**

   The web sites are built **using percentage** for the widths

2. **Media queries**

   Use a module from the CSS3 specification

3. **Flexible media & images**

   When screen size begins to change, the media/images need to be flexible to suit the screen size

TECHNIQUES FOR RWD: FLEXIBLE, GRID-BASED, LAYOUT

- Idea behind liquid layout: it's more carefully designed **in terms of proportion→ use percentage**
- Proportion of each page element is the target element divided by the context

  Example:
  - suppose your desktop layout has the main wrapper with the width of 960px and
  - suppose that the target element is 300px wide
  - then the proportion would be 31.25%

300px / 960px = 31.25%

TECHNIQUES FOR RWD: MEDIA QUERIES

- Media queries is the backbone of RWD
- Media queries provide the ability to
  - Specify different styles for individual browser device circumstances
  - Specify the width of the viewport or device orientation
- Using Media queries in the CSS file to change the styling of the HTML elements is based on certain breakpoints.

TECHNIQUES FOR RWD: FLEXIBLE MEDIA & IMAGES

- Using media queries, designers are able to:
  - Extend the media declarations to include various media properties, based on device being used. Such as:
    - screen size, orientation, and color
  - write a rule that prevents images from exceeding the width of their container

THE VIEWPORT META TAG

- Viewport meta tag:
  - Tells the browser how to behave when rendering the page – you tell the browser how big the viewport will be
  - Use the viewport meta tag in the <head> section
  - If we are using RWD, it's good to have the meta tag viewport as

```
<meta name="viewport"
content="width=device-width,
initial-scale=1">
```

No zooming

Adapt to the width of the device

CODING META VIEWPORT TAGS

- There are two ways to add the viewport tag for overriding the default viewport by user agent.
  1. Use the @viewport CSS rule.
     - This is still relatively new and mostly unsupported for now.

       ```
       /* CSS Document */
       @viewport {width: 480px; zoom: 1;}
       ```
  2. Use the viewport meta tag
     - This is almost supported universally.

       ```
       <meta name="viewport"
       content="width=device-width, initial-scale=1">
       ```

CODING META VIEWPORT TAGS (CONTINUED)

```
<meta name="viewport"
content="width=device-width,
initial-scale=1">
```

- width=device-width:
  - The page adapts to the device’s width
  - Syncs with the device’s width
- initial-scale=1:
  - Make the initial scale at 100%
  - When the viewport is larger than the screen width, the scale factor will shrink down to fit the width within the viewport.

CODING MEDIA QUERIES

- The following code will display the font-size at 100% if the width is at least 1024 px

```
@media screen and (min-width: 1024px) {
   body {font-size: 100%;}
}
```

- The following code tests the orientation and the device-width

```
@media screen and (min-device-width: 480px) and
(orientation: landscape) {
   body {  font-size: 100%;   }
   }
```

- The logical operators are pretty interchangeable:
  - The operator “and” can be replaced with “not”. The orientation “portrait” with “landscape”.

CODING MEDIA QUERIES (CONTINUED)

- The following code renders a page that the body background color will change to blue only between 500px and 700px.

```
@media screen (min-width:500px)and (Max-
width:700px){
  body {background: blue;}
  }
```

- The following code displays an orange background color when a device hits 1024px width and changes to yellow when the display of a device drop into mobile territory.

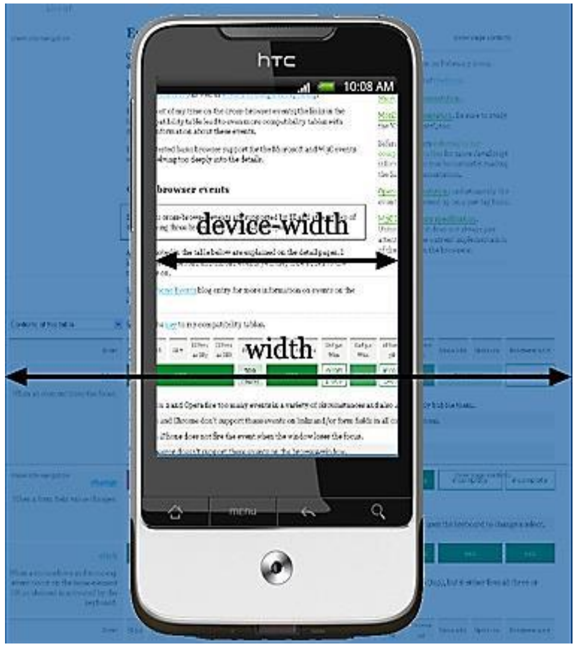
```
@media (max-width: 1024px) {
      body { background: orange;}
 }
 @media (max-width: 768px) {
    body {background: yellow;}
 }
```

DEFINITIONS

- Width = width of the display area
- Device-width = width of device
- Orientation = orientation of the device
- Aspect-ratio = ratio of width to height
  - It is expressed by two numbers separated by a slash
- Device-aspect-ratio = ratio of device-width to device-height
- Resolution – density of pixels of output device (dpi)

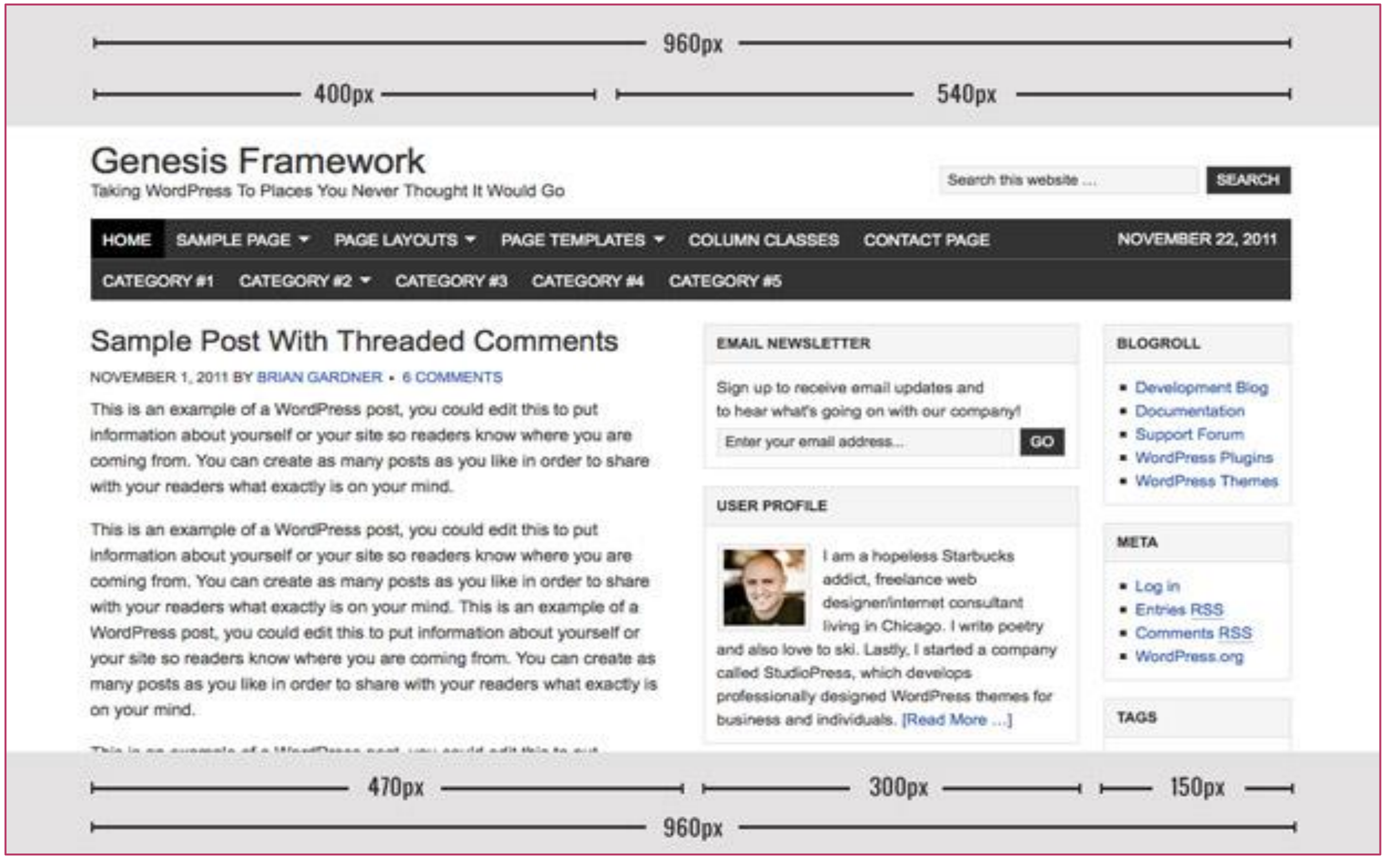
MEDIA QUERIES TOGETHER WITH VIEWPORT

- It is **<u>not</u>** a good idea to use the media queries without a meta viewport tag
- Some mobile browsers have a default layout viewport of around 850 to 1000 pixels
- The page will be much larger than the device width



CONVERTING AN EXISTING PAGE TO RWD

- Let’s say the existing page has the following layout



Genesis Framework
Taking WordPress To Places You Never Thought It Would Go

Search this website ... SEARCH

HOME SAMPLE PAGE PAGE LAYOUTS PAGE TEMPLATES COLUMN CLASSES CONTACT PAGE NOVEMBER 22, 2011

CATEGORY #1 CATEGORY #2 CATEGORY #3 CATEGORY #4 CATEGORY #5

Sample Post With Threaded Comments

NOVEMBER 1, 2011 BY BRIAN GARDNER • 6 COMMENTS

This is an example of a WordPress post, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many posts as you like in order to share with your readers what exactly is on your mind.

This is an example of a WordPress post, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many posts as you like in order to share with your readers what exactly is on your mind. This is an example of a WordPress post, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many posts as you like in order to share with your readers what exactly is on your mind.

EMAIL NEWSLETTER

Sign up to receive email updates and to hear what's going on with our company!

Enter your email address... GO

USER PROFILE

I am a hopeless Starbucks addict, freelance web designer/internet consultant living in Chicago. I write poetry and also love to ski. Lastly, I started a company called StudioPress, which develops professionally designed WordPress themes for business and individuals. [Read More ...]

BLOGROLL

- Development Blog
- Documentation
- Support Forum
- WordPress Plugins
- WordPress Themes

META

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

TAGS

CONVERTING AN EXISTING PAGE TO RWD (CONTINUED)

- Assume the existing page has the following basic structure of HTML code

```
<div id="wrap">
    <div id="header">
        <div id="title-area"></div>
        <div class="widget-area"></div>
    </div>
    <div id="inner">
        <div id="content-sidebar-wrap">
            <div id="content"></div>
            <div id="sidebar"></div>
        </div>
        <div id="sidebar-alt"></div>
    </div>
</div>
```

Converting an Existing page to RWD (continued)

- Assume the existing page has the following basic structure of CSS code

```
#wrap {width: 960px; }
#header {width: 960px;}
#title-area {width: 400px;}
#header .widget-area {width: 540px;}
#inner {width: 960px;}
#content-sidebar-wrap {width: 790px;}
#content {width: 470px;}
#sidebar {width: 300px;}
#sidebar-alt {width: 150px;}
```

Converting an existing page to RWD (continued)

- SUPPOSE THE TARGET GOAL IS 960PX WIDE

```
#wrap {width: 100%; }
#header {width: 100%;}
#title-area {width: 41.666667%;}
#header .widget-area {width: 56.25%;}
#inner {width: 100%;}
#content-sidebar-wrap {width: 82.291667%;}
#content {width: 48.958333%;}
#sidebar {width: 31.25%;}
#sidebar-alt {width: 15.625%;}
```

Formula:

(original pixels/target goal pixels)* 100%

```
Example for the #title-area:
            (400px/960px)*100% = 41.666667%
```

Converting an existing page to RWD (continued)

- The ul in the sidebar

```
/*The pixel for the margin is 25px */
   .widget-area ul {
       margin: 10px 0 0 25px;}

/*the percentage conversion of the
target margin*/
  .widget-area ul {
      margin: 10px 0 0 16.666667%;}
```

This pixel is 150 because that is the width of the sidebar.

(25/150) * 100%)= 16.666667%;

- Flexible images
  - img { max-width: 100%; }

CONVERTING EXERCISE: DO NOT ROUND UP!

Do not round up, keep the long decimal points

- Because each browser rounds the percentage differently, if you round the percentage, you need to tweak each section

Converting Exercise, Inserting Media Queries

- Add two media query break points at the end of the style section

  Note: The two media queries are provided for you at the right.

```
@media screen and (max-
width:830px) {
   #content {
      float: left;
      width: 98%;
      margin-top:5px;
   }
   nav li;
   nav a {
          display:block;
   }
}
@media screen and (max-
width:480px) {
   #content {
      float: none;
      width:95%;
   }
}
```

TESTING THE RESPONSIVE DESIGN

- Test with the new media queries to see whether or not they're hitting the right breakpoints.
  - Resize the browser window to see the changes
    - This is helpful and gives immediate feed back, however:
      - The feed back is not really the actual trigger points
      - It does not show how the site will render
      - It overlooks the performance

TESTING THE RESPONSIVE DESIGN (CONTINUED)

- Use online simulator testing tools
  - There are many free online testing tools to help test more precisely and to speed up the process.
- Using online mobile emulators:
  programs that simulate a specific mobile device, browser, or operating system
- Test on actual devices, best way, but it is expensive to have all the devices on hand and to purchase more new ones.

ONLINE SIMULATOR TESTING TOOLS

- Benjamin Keen Bookmarklet
  - http://www.benjaminkeen.com/open-source-projects/smaller-projects/responsive-design-bookmarklet/
- The following online simulator allows you to just enter the URL
  - Responsivepx by Remy Sharp: users have control of the precise width
    http://responsivepx.com/
  - Responsive.is: it provides icon for difference devices: http://www.headlondon.com/
  - Mobiltest: user can chose the devices, also provides the average load time

    http://mobitest.akamai.com/m/index.cgi

ONLINE EMULATOR TESTING TOOLS

- TestiPhone.com
- Opera's Mini simulator
- Download and install emulators:
  - Opera's Mobile emulator
  - Apple SDK, the emulators comes with Apple's iOS
  - Android SDK, the emulators comes with Android OS.

DEBUGGING TOOLS

- Tools for debugging when the behavoir is not expected after testing
  - Opera's Remote Debugger
    - Dragongly: Debug on the desktop with the site on a mobile device
  - WebKit remote debugging
    - Weinre
    - Web Inspector