SWE 363: Web Engineering & Development

Module 5

Responsive Web Design

Objectives

- ❑ Learn Responsive Web Designs

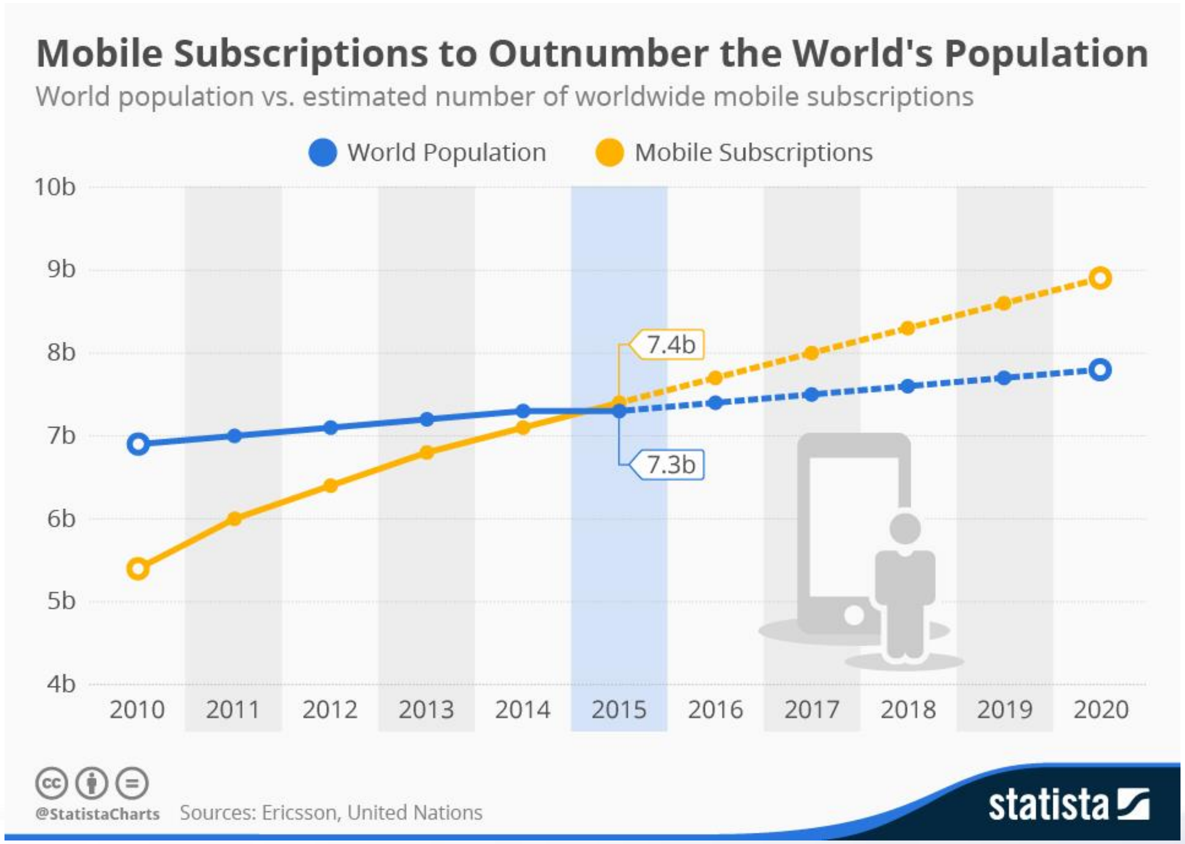
Outline

References

- "Fundamentals of Web Development" Book by Randy Connolly and Ricardo Hoar, 2015
- https://www.w3schools.com/ w3schools.com
- https://css-tricks.com/snippets/css/media-queries-for-standard-devices/
- http://www.free-css.com/template-categories/bootstrap

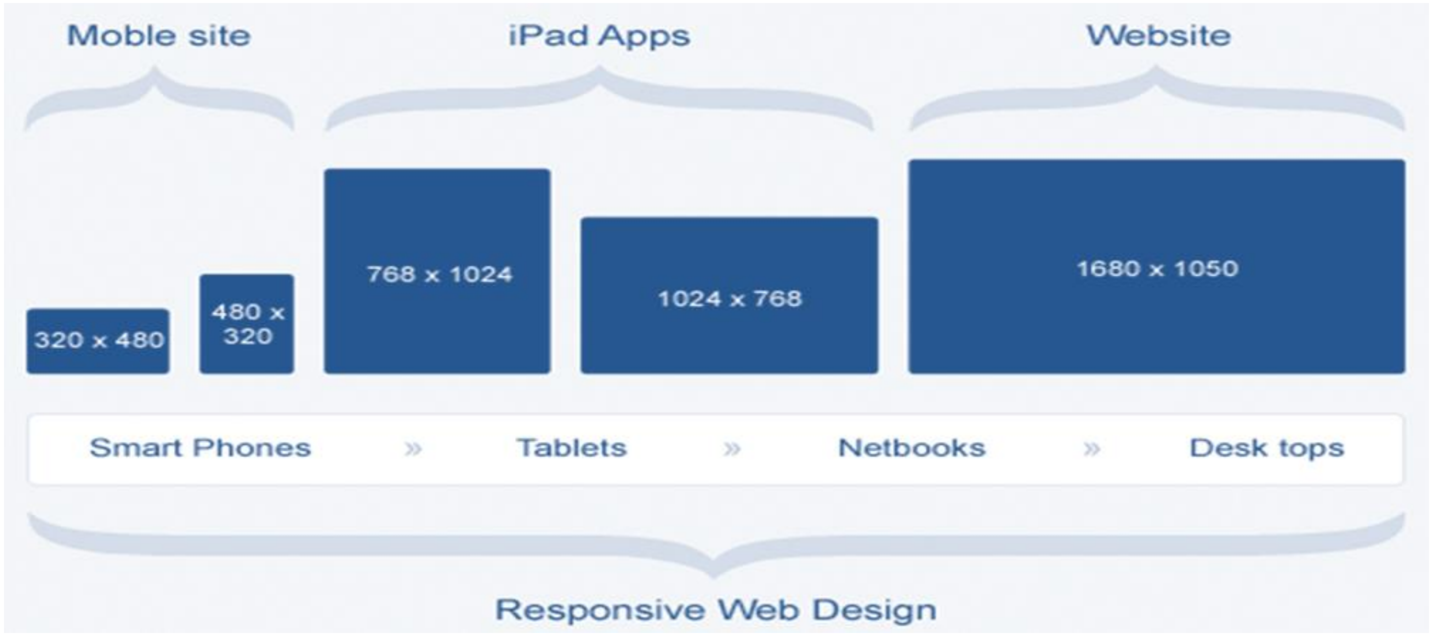
Introduction

- There are 6.8 billion people on the planet, 5.1 billion of whom own a cell phone.
- Starting from the year 2015 Mobile users accessing internet were more than computer users



Introduction

- Websites are viewed on a variety of devices beyond traditional desktops--from smartphones to tablets to game consoles to large-screen TVs to vehicles.



http://screensiz.es/

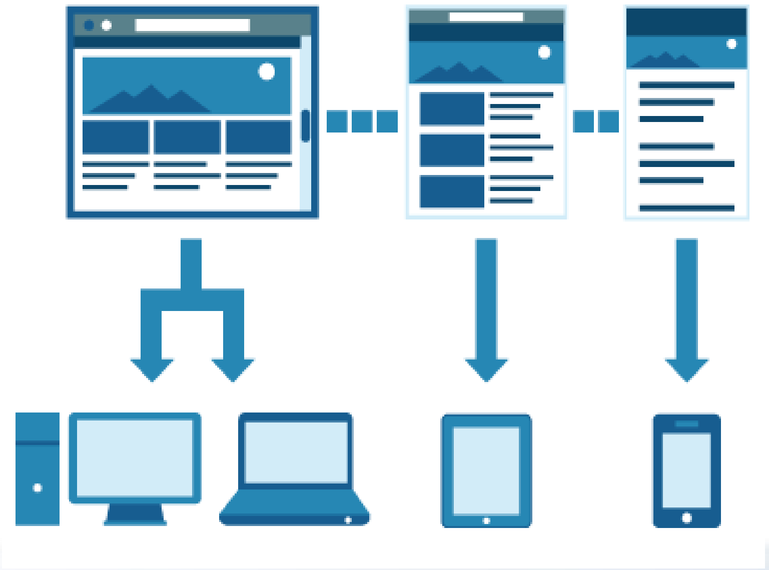
What is RWD?

- A Responsive Website serves the exact same page to every visitor but the design and layout of that page responds to the size of the visitors screen size.

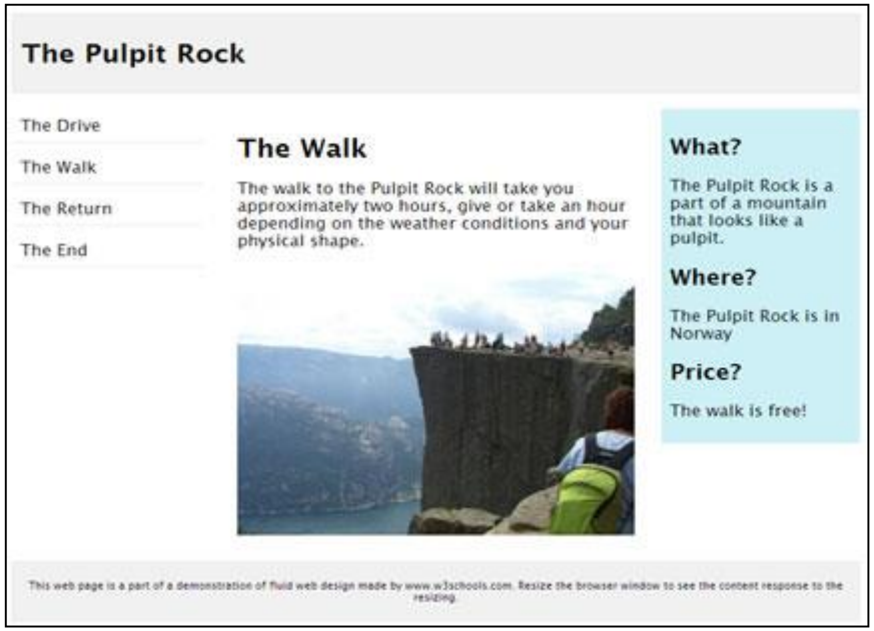
- RWD is a process that allows developer to design and develop a website that automatically adjust page view to look good on all devices
  - By adapting **content** and **layout** to fit the user screen without leaving out any information

- Goal: a web page should look good on **any device** (desktop, laptop, tablet, or phone)

- RWD uses only HTML and CSS and it is not a program or a JavaScript



Example



The Pulpit Rock

The Drive
The Walk
The Return
The End

The Walk

The walk to the Pulpit Rock will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.

What?

The Pulpit Rock is a part of a mountain that looks like a pulpit.

Where?

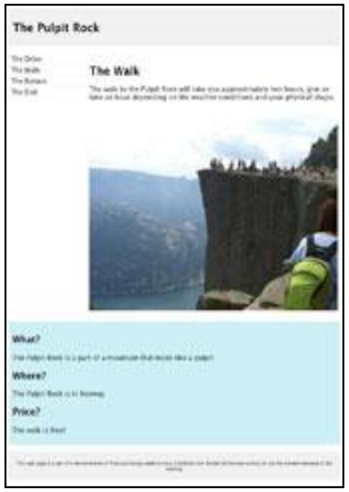
The Pulpit Rock is in Norway

Price?

The walk is free!

This web page is a part of a demonstration of fluid web design made by www.w3schools.com. Resize the browser window to see the content response to the resizing.

Desktop



The Pulpit Rock

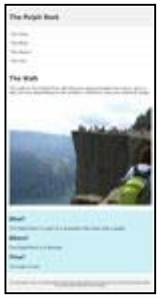
The Walk

What?

Where?

Price?

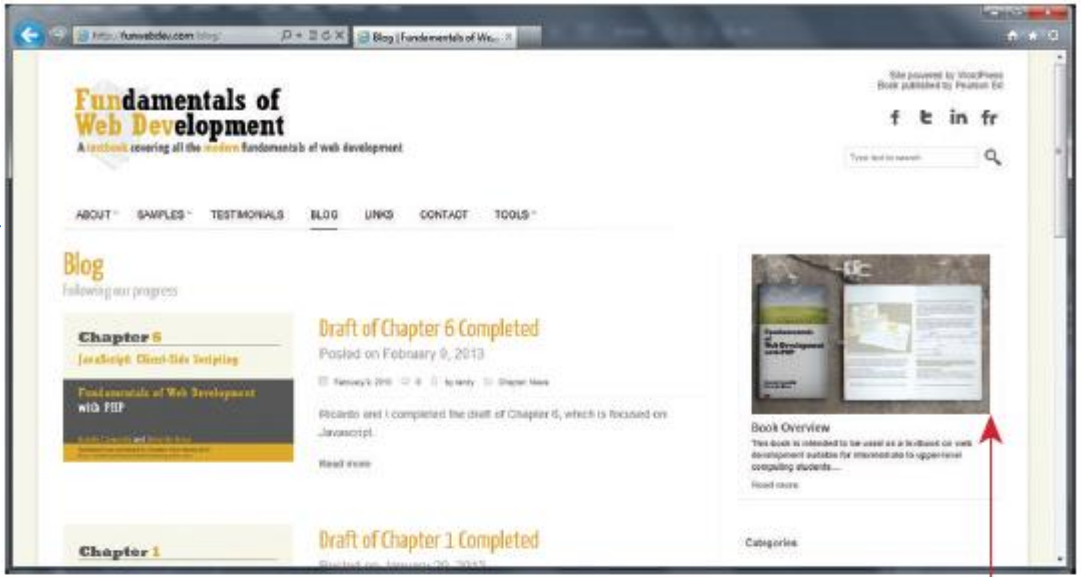
Tablet



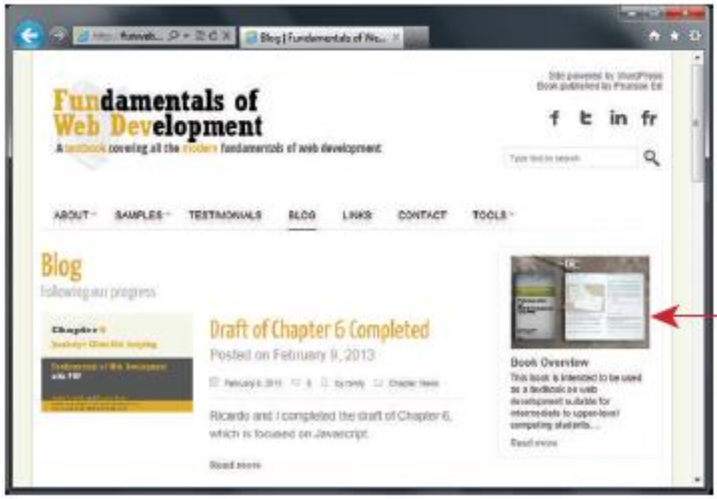
Phone

RWD uses CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen

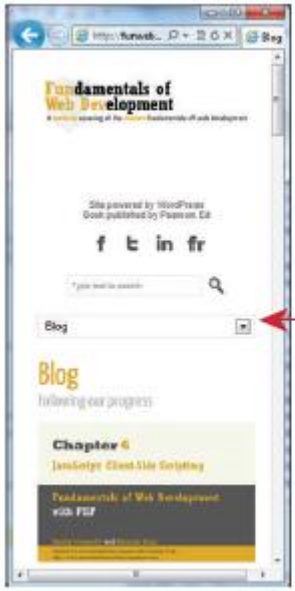
Try it



Fundamentals of Web Development
A textbook covering all the modern fundamentals of web development
ABOUT
SAMPLES
TESTIMONIALS
BLOG
LINKS
CONTACT
TOOLS
Blog
Following our progress
Chapter 6
JavaScript: Client-Side Scripting
Draft of Chapter 6 Completed
Posted on February 9, 2013
Ricardo and I completed the draft of Chapter 6, which is focused on Javascript.
Read more
Book Overview
Draft of Chapter 1 Completed
Categories



Notice how some elements are scaled to shrink as browser window reduces in size.



When browser shrinks below a certain threshold, then layout and navigation elements change as well.

In this case, the `<ul>` list of hyperlinks changes to a `<select>` and the two-column design changes to one column.

Why should we use RWD?

- Build once for all devices
  - Each year new devices are pouring into the market, RWD let us build one site, and modify it to adapt to the new device's screen size
- Easy to manage content editing through a single CMS (Content Management System)
- If we have a working website:
  - We can convert the existing it to a responsive Web site to adapt all kind of devices
  - No need to rebuild new websites to adapt to the new devices
- 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
  - We can keep the same SEO strategy

Comparing layouts

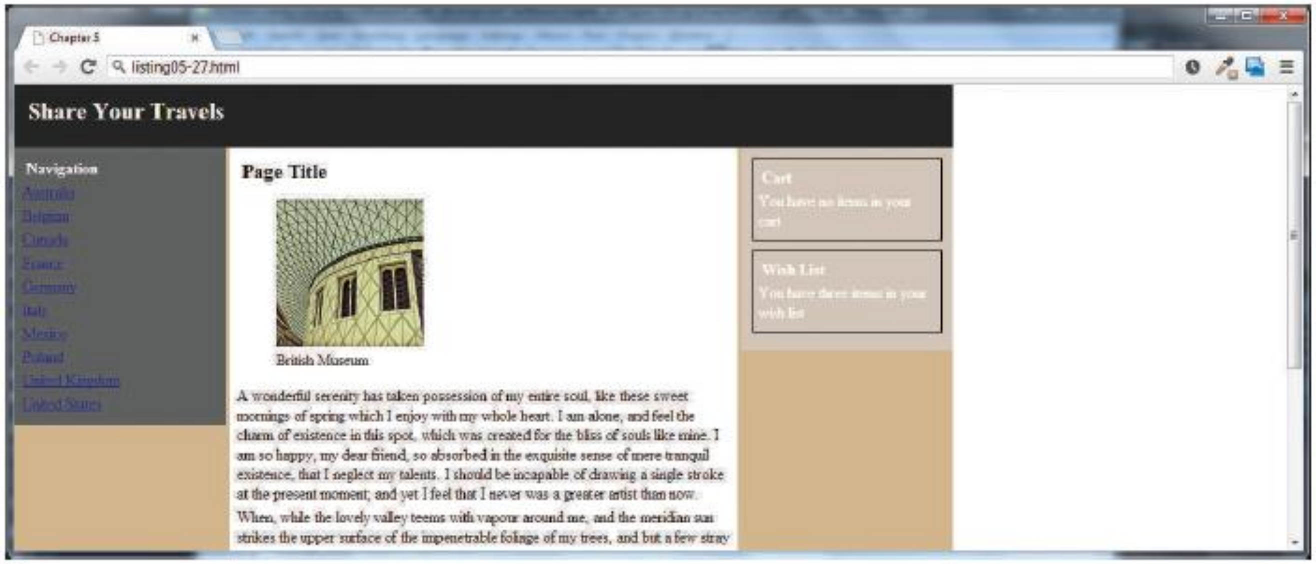
Fixed Layout

- In a fixed layout, the basic width of the design is set by the designer, typically corresponding to an "ideal" width based on a "typical" monitor resolution.

- Fixed layouts are created using pixel units, typically with the entire content within a *<div> container* (often named "container", "main", or "wrapper") whose width property has been set to some width.

- The advantage of a fixed layout is that
  - it is easier to produce
  - generally has a predictable visual result.
  - It is also optimized for typical desktop monitors;

  however, as more and more user visits are happening via smaller mobile devices, this advantage might now seem to some as a disadvantage.



Share Your Travels

Navigation
Australia
Belgium
Canada
France
Germany
Italy
Mexico
Poland
United Kingdom
United States

Page Title

British Museum

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.

When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper surface of the impenetrable foliage of my trees, and but a few stray

Cart
You have no items in your cart

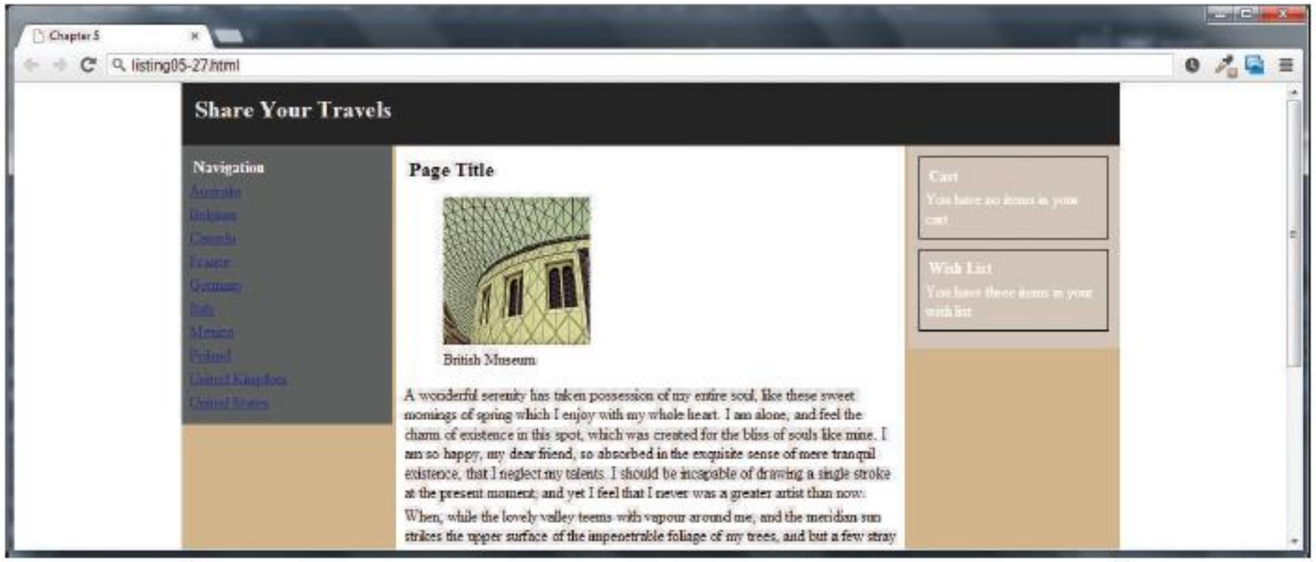
Wish List
You have three items in your wish list

960px

Extra space to right

```
div#wrapper {
  width: 960px;
  background-color: tan;
}
```

```
<body>
   <div id="wrapper">
      <header>
      ...
      </header>
      <div id="main">
      ...
      </div>
      <footer>
      ...
      </footer>
   </div>
</body>
```



Share Your Travels

Navigation
Australia
Belgium
Canada
France
Germany
Italy
Mexico
Poland
United Kingdom
United States

Page Title

British Museum

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.

When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper surface of the impenetrable foliage of my trees, and but a few stray

Cart
You have no items in your cart

Wish List
You have three items in your wish list

960px

Equal space to the left and to right

```
div#wrapper {
  width: 960px;
  margin-left: auto;
  margin-right: auto;
  background-color: tan;
}
```

"*Fundamentals of Web Development*" Book by Randy Connolly and Ricardo Hoar, 2015

Liquid Layout

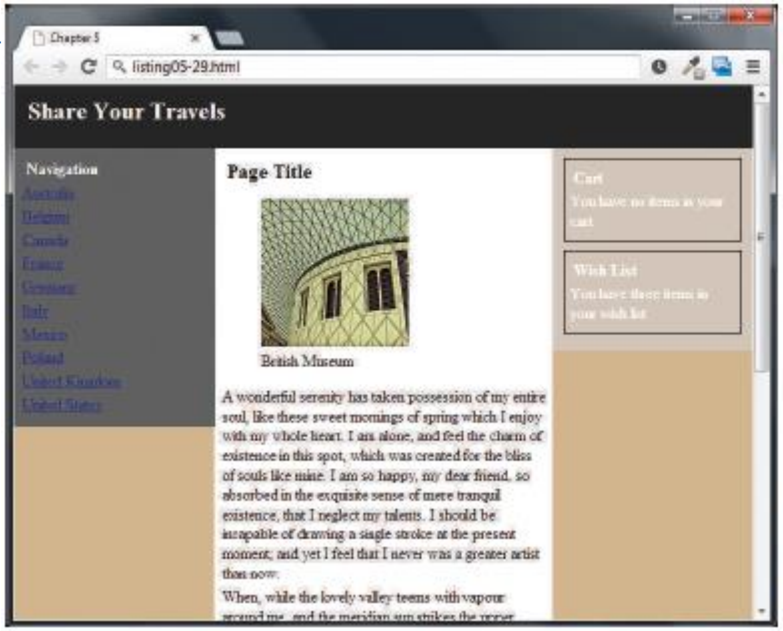
- Another approach to dealing with the problem of multiple screen sizes.
- In Fluid or Liquid Layout, define dimensions in percentages values rather than pixels.
- Percentages alone will not accommodate a wide variety of devices.
- To accommodate varying dimensions, fluid layouts need to remain simple.
- The percentage values in CSS are a percentage of the current browser width, so a layout in which all widths are expressed as percentages should adapt to any browser size.
- *The obvious advantage* of a liquid layout is that
  - it adapts to different browser sizes, so there is neither wasted white space nor any need for horizontal scrolling.

Liquid Layout

However,

- Images, video and other types of content with set widths <u>may need to be set at multiple widths</u> to accommodate different screen resolutions.

- With incredibly large screen resolutions, a lack of content may create <u>excess white space</u> that can diminish aesthetic appeal.



Chapter 5

listing05-29.html

Share Your Travels

Navigation
Australia
Belgium
Canada
France
Germany
Italy
Mexico
Poland
United Kingdom
United States

Page Title

British Museum

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.

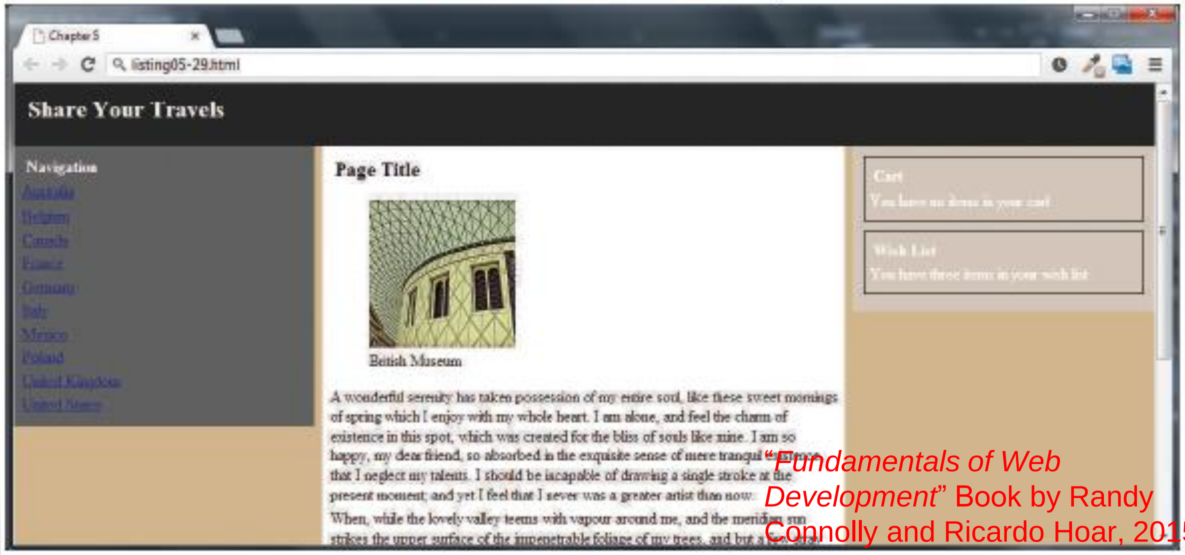
When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper

Cart
You have no items in your cart

Wish List
You have three items in your wish list

Fluid layouts are based on the browser window.

However, elements can get too spread out as browser expands.



Chapter 5

listing05-29.html

Share Your Travels

Navigation
Australia
Belgium
Canada
France
Germany
Italy
Mexico
Poland
United Kingdom
United States

Page Title

British Museum

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.

When, while the lovely valley teems with vapour around me, and the meridian sun strikes the upper surface of the impenetrable foliage of my trees, and but a few stray

Cart
You have no items in your cart

Wish List
You have three items in your wish list

"*Fundamentals of Web Development*" Book by Randy Connolly and Ricardo Hoar, 2015

Key Components of RWD

- There are four key components that make responsive design work

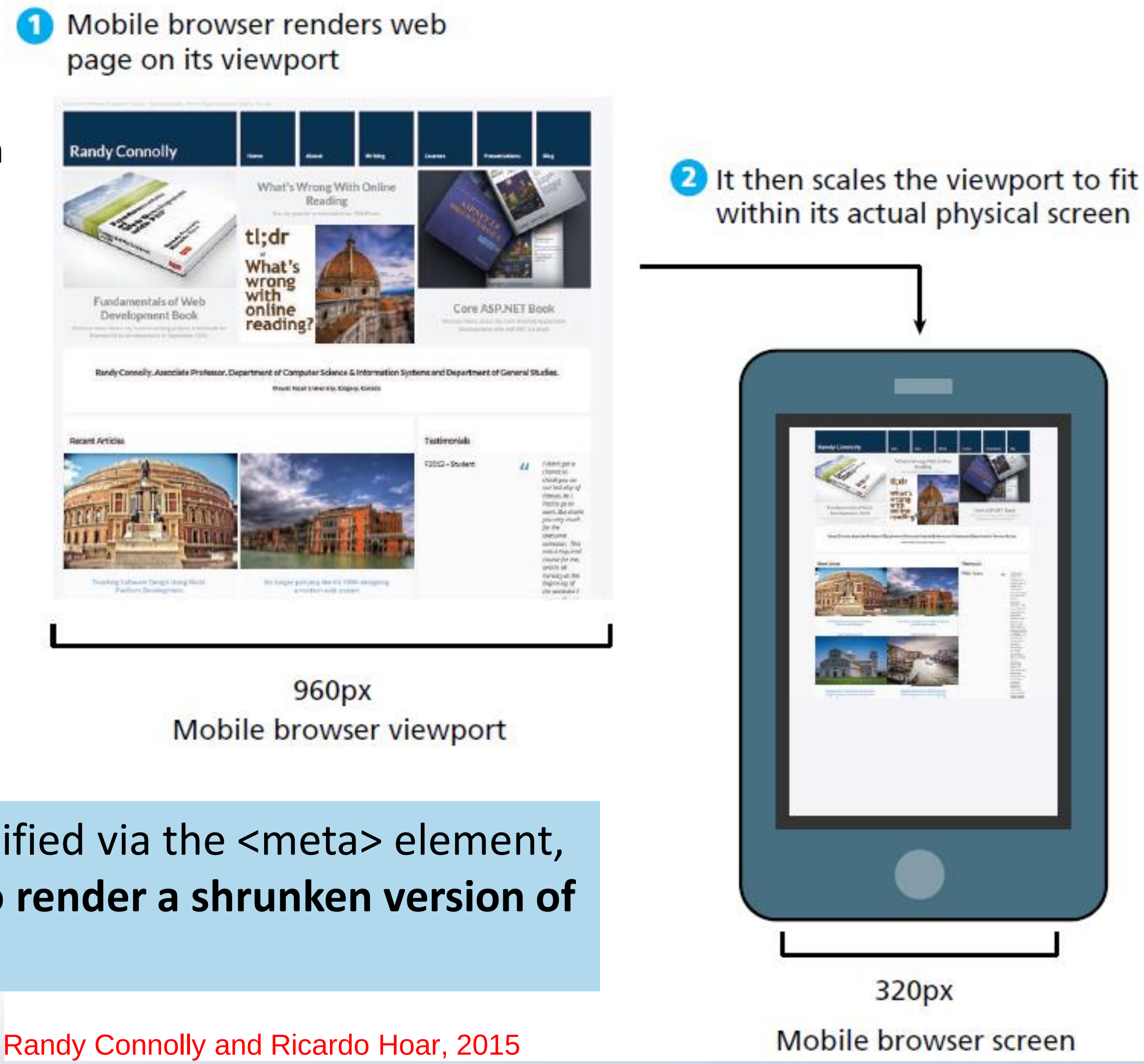
  1. Liquid layouts
  2. Scaling images to the viewport size
  3. Setting viewports via the <meta> tag
  4. Customizing the CSS for different viewports using media queries

- Responsive designs begin with a liquid layout,
  - most elements have their widths specified as percentages
  - To make images scale in size, specify the following rule:
    - Resizing image to fit the screen resolution.
    - Hiding non-essential elements especially for smaller screen.
    - Optimize your page for vertical scrolling.

Viewports

Setting Viewports

- The viewport is the user's visible area of a web page.
  - The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.



1 Mobile browser renders web page on its viewport
2 It then scales the viewport to fit within its actual physical screen
960px
Mobile browser viewport
320px
Mobile browser screen

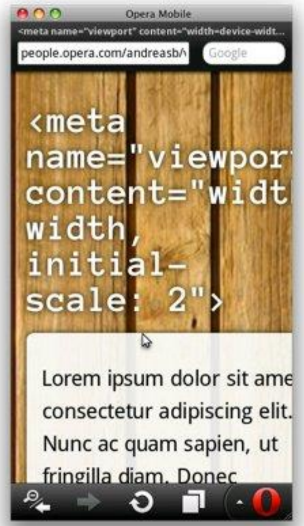
if an alternate viewport is not specified via the <meta> element, then the mobile browser will try to **render a shrunken version of the full desktop site.**

Setting Viewports…

❑ HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag and should be included in all web pages

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

- A <meta> viewport element:
  - ➢ gives the browser instructions on how to control the page's dimensions and scaling.
- width=device-width:
  - ➢ set the page width to follow the screen-width of the device
- initial-scale=1:
  - ➢ Make the initial scale at 100%
  - ➢ sets the initial zoom level when the page is first loaded by the browser.



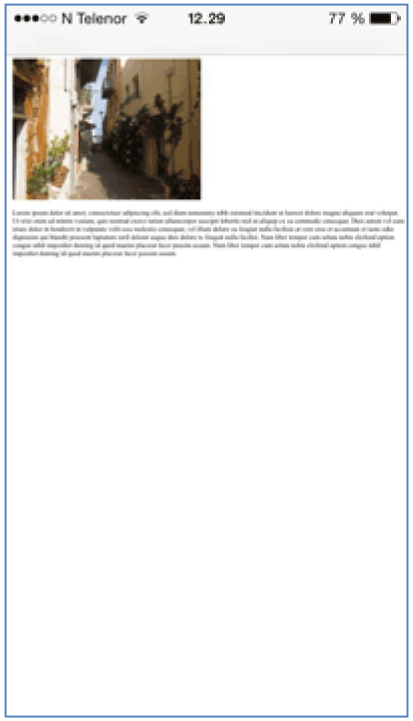
<meta name="viewport" content="width=device-widt…

<meta
name="viewpor
content="widt
width,
initial-
scale: 2">

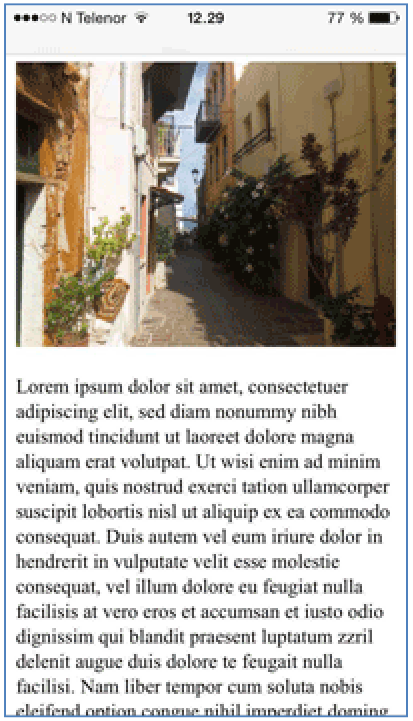
Lorem ipsum dolor sit ame
consectetur adipiscing elit.
Nunc ac quam sapien, ut
fringilla diam. Donec

https://www.w3schools.com/css/css_rwd_viewport.asp

With & without using viewport meta tag: An Example



[illegible]



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

[Without the viewport meta tag]

[With the viewport meta tag]

Setting Viewports..

- ❑ There is another way to add the viewport tag for overriding the default viewport by user agent

Using the @viewport CSS rule.

```
/* CSS Document */
@viewport {width: 480px; zoom: 1;}
```

- ❑ This is still relatively new and mostly unsupported for now.

Media Queries

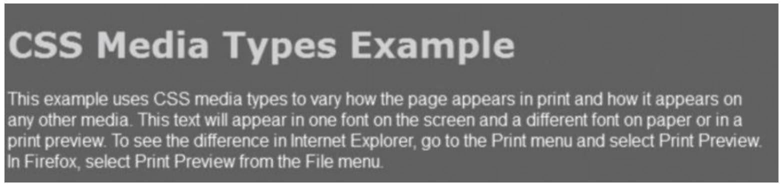
Media Queries

- Media queries is the backbone of RWD
- A media query is a way to apply style rules based on the medium that is displaying the file.
  - These queries are Boolean expressions and can be added to your CSS files
- Using Media queries in the CSS file to change the styling of the HTML elements is based on certain breakpoints.
- Media queries can change styles to match the device size, screen type and orientation (portrait and landscape).
- The main purpose of a media query is to apply different CSS rules in order to obtain different layouts, depending on the width of the display window afforded to your content.

example

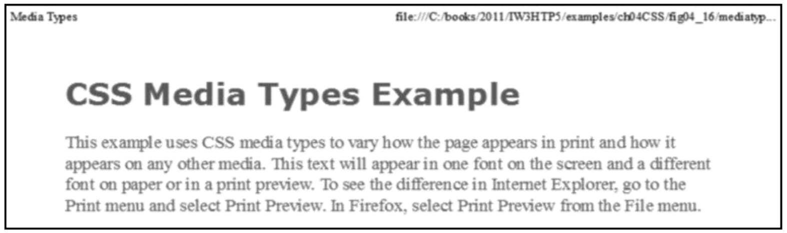
A simple classic example that applies one set of styles when the document is viewed on all media (including screens) other than a printer, and another when the document is printed.

```
@media screen{
body     {background: #ff0000;}
h1{ color: grey;}
p {font-size:12pt; color: white;}
}


@media print {
body     {background: #fff;}
h1{ color: black;}
p {font-size:11pt; color: black;}

}
```



CSS Media Types Example

This example uses CSS media types to vary how the page appears in print and how it appears on any other media. This text will appear in one font on the screen and a different font on paper or in a print preview. To see the difference in Internet Explorer, go to the Print menu and select Print Preview. In Firefox, select Print Preview from the File menu.



Media Types file:///C:/books/2011/IW3HTP5/examples/ch04CSS/fig04_16/mediatyp...

CSS Media Types Example

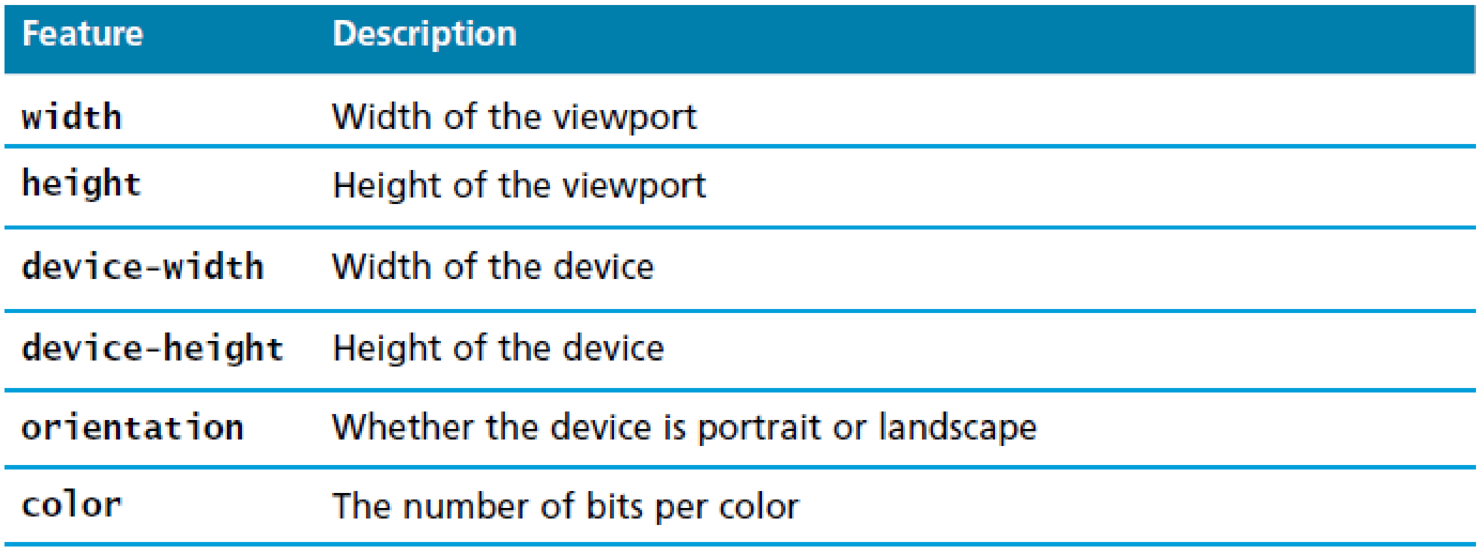
This example uses CSS media types to vary how the page appears in print and how it appears on any other media. This text will appear in one font on the screen and a different font on paper or in a print preview. To see the difference in Internet Explorer, go to the Print menu and select Print Preview. In Firefox, select Print Preview from the File menu.

Most common media type for a web page is the `screen` media type, which is a standard computer screen

Media Queries

❑ Media Queries

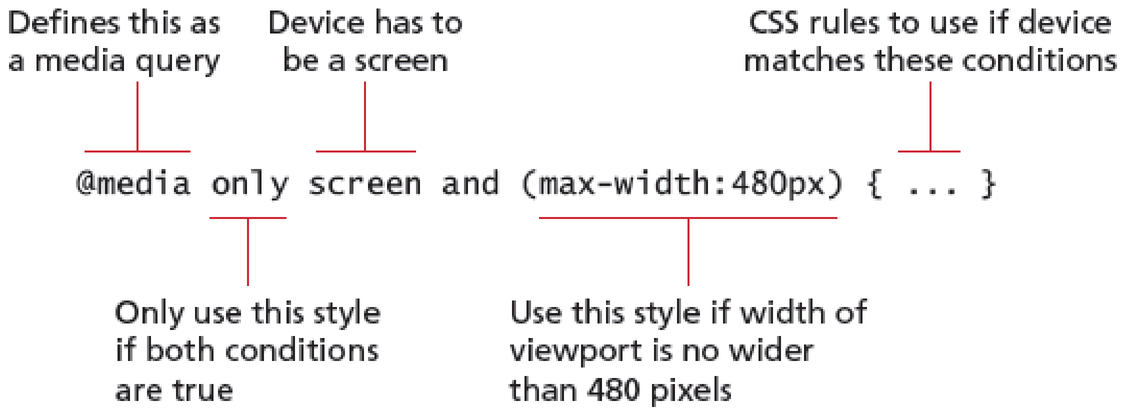
- Allow you to format your content to specific output devices.
- Include a media type and expressions that check the media features of the output device.
- Common media features include:



Feature	Description
`width`	Width of the viewport
`height`	Height of the viewport
`device-width`	Width of the device
`device-height`	Height of the device
`orientation`	Whether the device is portrait or landscape
`color`	The number of bits per color

Media Queries..

- The most commonly used media feature is width.


Defines this as a media query
Device has to be a screen
CSS rules to use if device matches these conditions
@media only screen and (max-width:480px) { ... }
Only use this style if both conditions are true
Use this style if width of viewport is no wider than 480 pixels

```
@media screen and (min-width: 480px) {
    .content { float: left; }
    .social_icons { display: none }
    // and so on...
}
```

Media Queries..

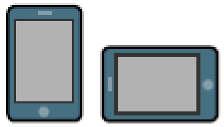
- Modern responsive sites will typically provide CSS rules for
  - phone displays first,
  - then tablets,
  - then desktop monitors,

  This approach called *progressive enhancement*, in which a design is adapted to progressively more advanced devices

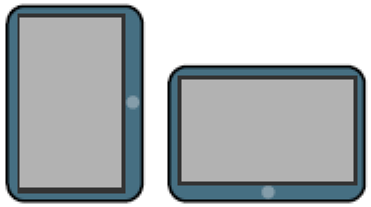
Use the Mobile First Approach and Progressive Enhancement

- How a responsive site might use media queries to provide progressive enhancement.
  - the smallest device is described first, while the largest device is described last. Since later (in the source code) rules override earlier rules, this provides progressive enhancement, meaning that as the device grows you can have CSS rules that take advantage of the larger space.

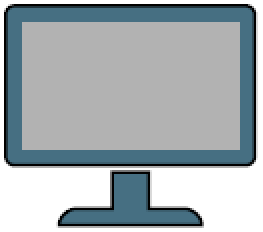
styles.css



```
/* rules for phones */
@media only screen and (max-width:480px)
{
  #slider-image { max-width: 100%; }
  #flash-ad { display: none; }
   ...
}
```



```
/* CSS rules for tablets */
@media only screen and (min-width: 481px)
     and (max-width: 768px)
{
   ...
}
```



```
/* CSS rules for desktops */
@media only screen and (min-width: 769px)
{
   ...
}
```

Instead of having all the rules in a single file, we can put them in separate files and add media queries to `<link>` elements.

```
<link rel="stylesheet" href="mobile.css"  media="screen and (max-width:480px)" />
<link rel="stylesheet" href="tablet.css"  media="screen and (min-width:481px)
    and (max-width:768px)" />
<link rel="stylesheet" href="desktop.css" media="screen and (min-width:769px)" />
```

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..

- ❑ 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;}
 }
```

Typical Device Breakpoints

```css
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Media Queries: Examples

https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries1

https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries2

https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_image3

https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_breakpoints

https://css-tricks.com/snippets/css/media-queries-for-standard-devices/

Responsive Images

RWD: Images

- If the width property is set to 100%, the image will be responsive and scale up and down:

```
img {
    width: 100%;
    height: auto;
}
```

Allows images to be scaled up to be larger than original size

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

max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size

https://www.w3schools.com/css/css_rwd_images.asp

RWD: Background Images

- Background images can also respond to resizing and scaling by one of the following methods
  1. Set background-size property to "contain"
     - But will keep the image aspect ratio

```
div {
    width: 100%;
    height: 400px;
    background-image: url('img_flowers.jpg');
    background-repeat: no-repeat;
    background-size: contain;
    border: 1px solid red;
}
```

  2. Set background-size property to "100% 100%” to stretch the image to cover the entire content area

  3. Set background-size property to “cover” to scale the image to cover entire content area.

https://www.w3schools.com/cssref/playit.asp?filename=playcss_background-size&preval=cover

Adjusting background for a Mobile Device

Takes less time to load small images

```
/* For width 481px and larger: */
#logo {
       background: url(images/logo.png);
       width: 600px;
       border: 1px #ccc solid;
}

/* For width smaller than 481px: */
@media only screen and (max-device-width: 480px) {
       #logo {
              background: url(images/logo_mobile.png);
              width: 440px;
       }
}
```

Adjust logo for small viewport

Different Images for Different Devices

- <picture> element works similar to the <video> and <audio> elements to set up different sources, and the first source that fits the preferences is the one being used:

```
<picture>

  <source srcset="img_smallflower.jpg" media="(max-width: 400px)">
  <source srcset="img_flowers.jpg" media="(min-width: 465px)">
  <img src="img_flowers.jpg" alt="Flowers">

</picture>
```

https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_picture

RWE- Frameworks

RWD: Frameworks

- There are many existing CSS Frameworks that offer Responsive Design.
- They are free, and easy to use.

- Frontend frameworks usually consist of a package made up of a structure of files and folders of standardized code (HTML, CSS, JS documents etc.)

- The usual components are:
  - CSS source code to create a grid: this allows the developer to position the different elements that make up the site design in a simple and versatile fashion.
  - Typography style definitions for HTML elements.
  - Solutions for cases of browser incompatibility so the site displays correctly in all browsers.
  - Creation of standard CSS classes which can be used to style advanced components of the user interface.

RWD: Frameworks

- Many free and easy to use CSS Frameworks that offer RWD, e.g.
  - W3.CSS
    - https://www.w3schools.com/w3css/default.asp
  - Twitter Bootstrap (uses HTML, CSS, jQuery)
    - Developed by Mark Otto and Jacob Thornton at Twitter
    - Released as an open source product in August 2011 on GitHub
    - In June 2014 Bootstrap was the No.1 project on GitHub!
    - https://www.w3schools.com/bootstrap/default.asp
  - Foundation
  - Pure
  - Skeleton
  - Montage
  - Siimple
  - Gumby
  - Semantic UI
  - Cascade
  - CreateJS
  - Zebra

https://blog.templatetoaster.com/best-responsive-web-design-frameworks/

https://www.webpagefx.com/blog/web-design/html5-frameworks/

What is Bootstrap?

- Free front-end framework for faster and easier web development
- Includes HTML + CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Enables you to easily create responsive designs

https://www.w3schools.com/bootstrap/bootstrap_get_started.asp

Advantages of Bootstrap

- Easy to use:
  - Can be used by anybody with basic knowledge of HTML and CSS
- Responsive CSS features
  - To adjust to phones, tablets, and desktops
- Mobile-first approach
  - The first step in the design and implementation of a new website should be design and development of its **mobile version to support the increasing audience/users**
  - In Bootstrap, mobile-first styles are part of the core framework
- Compatible with all modern browsers
  - e.g. Chrome, Firefox, Internet Explorer, Safari, and Opera

Using Bootstrap

- There are two ways to start using Bootstrap
  - Download and install Bootstrap from getbootstrap.com
    - https://getbootstrap.com/docs/3.3/getting-started/
    - Gets zip file containing style sheets, fonts and javascript code
  - Include Bootstrap from a CDN

    https://www.w3schools.com/bootstrap/bootstrap_get_started.asp

- Bootstrap Tutorial

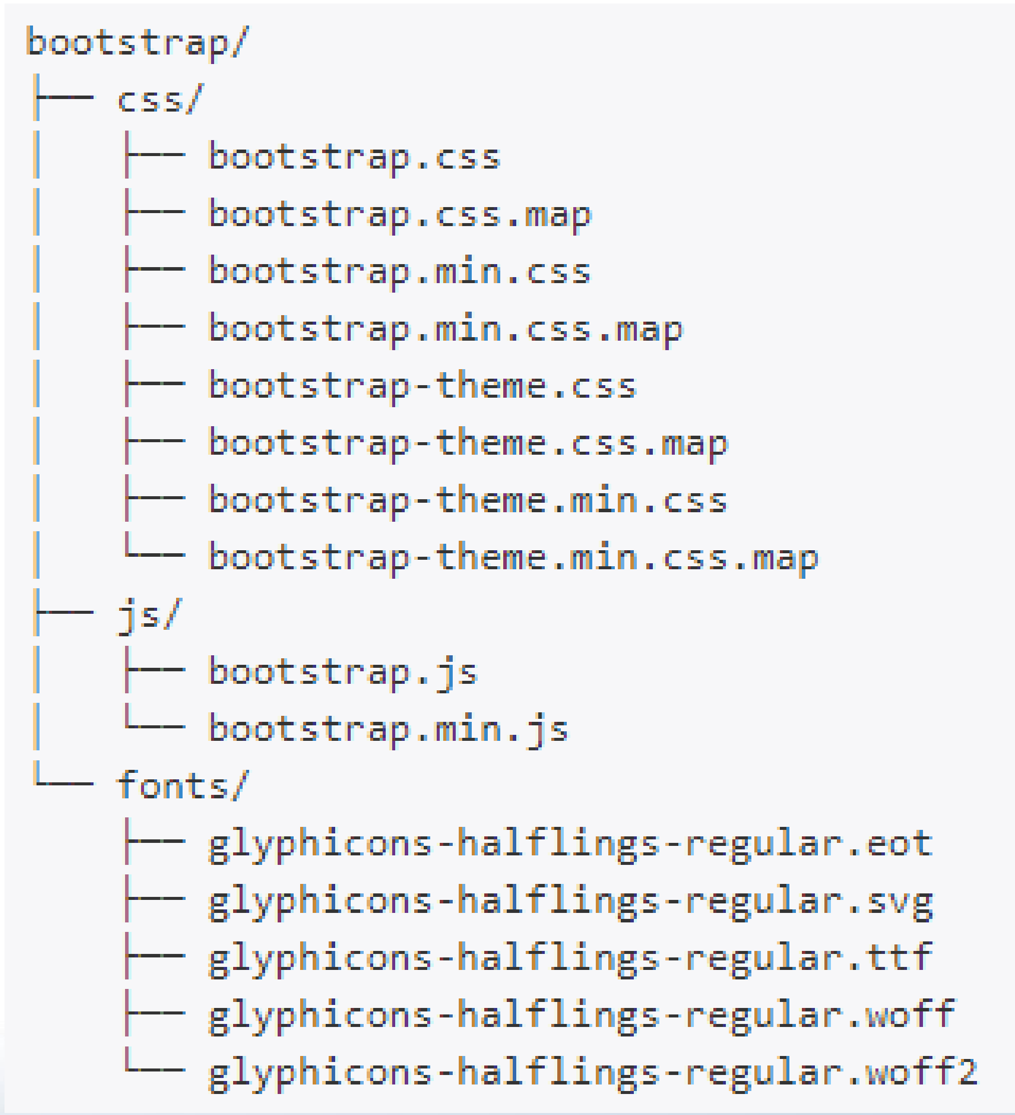
  https://www.youtube.com/watch?v=aTLRdrRQyN4

File Structure

▶PRECOMPILED BOOTSTRAP

- Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure:

- There are compiled CSS and JS (bootstrap.*), as well as compiled and minified CSS and JS (bootstrap.min.*). Fonts from Glyphicons are included, as is the optional Bootstrap theme.



```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

Example

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these
tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

https://getbootstrap.com/docs/3.3/getting-started/#template

Bootstrap Grid System

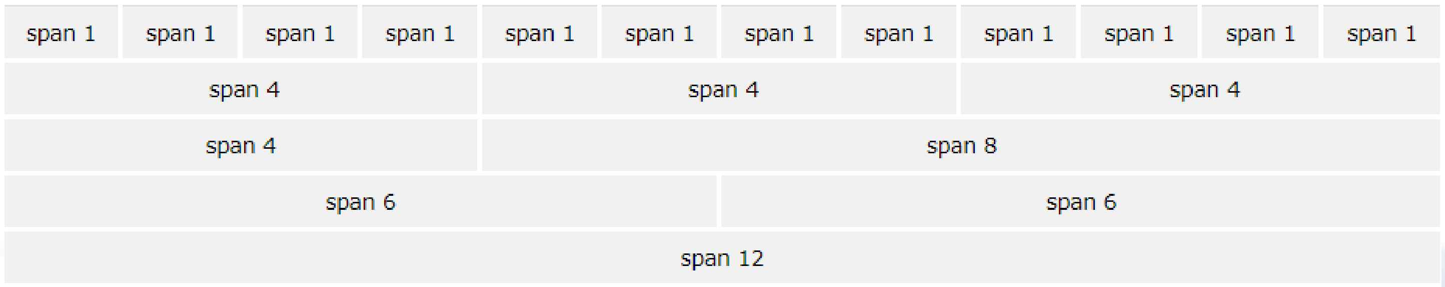
- Bootstrap grid system consists of
  - Container (to hold rows and columns)
  - Columns (up to 12 columns); can be grouped to create wider columns
  - Rows (unlimited)

- Container
  - Fixed container (has fixed width)

  <div class = “container”> …… </div>

  - Fluid container (takes the whole width of the screen)

  <div class = “container-fluid”> …… </div>



Bootstrap Grid System

- BS grid system has four classes:
  - xs (for phones - screens < 768px wide)
  - sm (for tablets - screens >= 768px wide)
  - md (for small laptops - screens >= 992px wide)
  - lg (for laptops and desktops - screens >= 1200px wide)

- Example

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
</div>
```

https://www.w3schools.com/bootstrap/tryit.asp?filename=trybs_grid_ex1&stacked=h

https://www.w3schools.com/bootstrap/bootstrap_grid_stacked_to_horizontal.asp

BS grid system rules

- Use rows to create horizontal groups of columns
- Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like .row and .col-sm-4 are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding
- Grid columns are created by specifying the number of 12 available columns you wish to span, e.g. three equal columns would use three .col-sm-4
- Column widths are in percentage, so they are always fluid and sized relative to their parent element

Bootstrap Studio

https://bootstrapstudio.io/

15 powerful Bootstrap Studio features that will help you save time and be more productive:

https://www.youtube.com/watch?time_continue=292&v=KSJ7Kfonn7I

- ❑ Component drag & drop
- ❑ Selection and component options
- ❑ Realtime preview
- ❑ Responsive tools
- ❑ Beautiful HTML
- ❑ Built-in themes
- ❑ Integrated icon fonts
- ❑ Multi-page designs
- ❑ Linked components
- ❑ Custom components
- ❑ Importing assets
- ❑ Writing HTML
- ❑ Smart CSS editing
- ❑ JavaScript editing
- ❑ Link external resources

Responsive Wed Design Online Resources

- ❑ Creating a Mobile-First Responsive Web Design
  http://www.html5rocks.com/en/mobile/responsivedesign/
- ❑ How Fluid Grids Work in Responsive Web Design
  http://www.1stwebdesigner.com/tutorials/fluid-grids-in-responsive-design/
- ❑ Responsive Web Design Techniques, Tools and Design Strategies
  http://mobile.smashingmagazine.com/2011/07/22/responsive-web-design-techniques-tools-and-design-strategies/
- ❑ Good information about the viewport meta tag
  http://www.paulund.co.uk/understanding-the-viewport-meta-tag
- ❑ The Ultimate Responsive Web Design Beginners Resource List
  http://www.targetlocal.co.uk/responsive-web-design-resources/
- ❑ Developing Mobile Applications: Web, Native, or Hybrid?
  https://blogs.oracle.com/fusionmiddleware/entry/developer_s_corner_developing_mobile

Online Simulator Testing Tools

- The following online simulator allows you to just enter the URL
  - http://www.responsinator.com/
  - 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