

HTML & CSS TRAINING FOR WEB DEVELOPMENT

MONIKA ORTEGA

Web & Graphic Designer / Web Developer
2000 – PRESENT (16 Years)



/MONIKAORTEGACOM



ph.linkedin.com/in/MONIKAORTEGA

TOPICS

DAY 1

Introduction of HTML
Structure
Tags & Content
HTML5

DAY 2

CSS (Styling)
Box Layout w/ HTML5
Create menu
Project Site – Wireframes

DAY 3

Creating DD menu
Project Site
Bootstrapping
Other HTML5/CSS tags
Uploading

All structure and content

Adding color & design

Finishing project website

TERMS

- Web Page
- Browser
- Web Server
- Tags
- URL
- Editor
- Nested

SUBLIME

- Download Sublime Editor at
<http://www.sublimetext.com/>
- Create HTML files: reference.html, template.html
- **USEFUL SHORTCUTS**
 - Ctrl + C = Copy
 - Ctrl + X = Cut
 - Ctrl + Shift + Arrow Up/Down = Move line up/down
 - Ctrl +] = indent line
 - Ctrl + [= unindent
 - Ctrl + Shift + D = Duplicate

HTML

- Hyper Text Markup Language
- Defines the content of a webpage
- .html or .htm file extension



HTML SYNTAX

ELEMENT > ATTRIBUTE > VALUE

SYNTAX:

```
<p title="text">paragraph</p>
```

Start tag

Attribute

Value

End tag

ELEMENTS = come in pairs

ATTRIBUTE = additional info placed in the start tag some have values, some don't

EMPTY HTML ELEMENTS

Empty HTML elements - elements with no content

-

-
- <meta />
- <hr>
- <input />

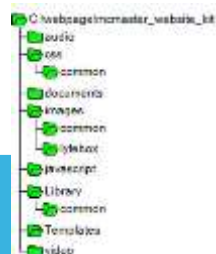
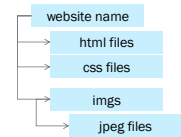
GLOBAL ATTRIBUTES

Can be used on ANY element

- title** = tooltip for an element
- style** = inline CSS
- class** = to assign 1 or more class(es) to an element
- id** = to assign a unique identifier to an element

FILING / DIRECTORIES

- Use **subfolders** for images, documents, media, etc.
- Use natural language to name folders and files.
- Typeset all file and folder names in **lowercase**.
- Only use the lowercase characters [a-z], the numbers [0-9] and the hyphen character [-] in folder and file names.
- Use a hyphen/dash between words.
- Use standard file extensions.
- Use prefixes and suffixes to help identify and group related files.



EXERCISE: Create your own root directory for files.

BASIC HTML STRUCTURE TAGS

All HTML documents contains the following tags for structure:

- <HTML> ... </HTML>
- <HEAD> ... </HEAD>
- <TITLE> ... </TITLE>
- <BODY> ... </BODY>

*all HTML docs are nested



Exercise:

Create basic structure for reference.html & template.html

<HTML> TAG

- **!DOCTYPE**
 - Rules/version for the markup language
 - Before <html> tag
 - Not case sensitive

```
<!DOCTYPE html>
<html>
  <head>
    <title> ... </title>
  </head>
  <body> ... </body>
</html>
```

HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

HTML 5

```
<!DOCTYPE html>
```

Exercise:

Put !Doctype in reference.html & template.html

<HEAD> TAG

- **<STYLE></STYLE>**

- **<META>**

(`<meta> cont...`)

```
<head>
  <title>Site Title</title>
  <style> </style>
  <meta name="keywords">

</head>
```

<META> TAG

- Character set
- Name
- Description
- Keywords
- Author

Exercise:

reference.html &
template.html

```
<head>
<meta charset="UTF-8">
<meta name="description"
content="Personal site">

<meta name="keywords"
content="Profile, pictures,
email">

<meta name="author"
content="Your Name">
</head>
```

<META> TAG

- **Name** → Lodestar Engineering (Philippines) - Official Website
- **Description** → Lodestar Engineering Supply Corp. (owner of Lodestar Philippines) and Solar products in the Philippines for over 30 years.
- **Keywords** → Lodestar Engineering (Philippines) - Official Website
Lodestar Engineering Supply Corp. (owner of Lodestar Philippines) and Solar products in the Philippines for over 30 years.
Lodestar Engineering Supply Corp. (owner of Lodestar Philippines) and Solar products in the Philippines for over 30 years.
Solar Products - Lodestar Engineering Philippines

REVIEW

- Basic structure of an HTML page
- What are the purpose of each basic structure tag?
- What is !doctype for?
- What are meta tags for?

TEXT ELEMENTS

- **Headers** `<h1> </h1>` (1-6)
- **Paragraph** `<p> </p>`
- **Break** `
`
- **Link** `<a> `

*links will be discussed in detail later
*smaller = less imp.

Exercise:

reference.html

```
<body>
  <h1>Heading</h1>

  <p>Lorem ipsum dolor
  <br> sit amet.</p>

  <a href="url">link</a>

</body>
```

TEXT ELEMENTS

- **Bold** ` `
- *Italicized* `<i> </i>`
- Underline `<u> </u>`
- Small `<small> </small>`
- *Emphasized* ` `

```
<p>
  <b>Bold</b>
  <i>Italicized</i>
  <u>Underline</u>
  <small>Small</small>
  <em>Emphasized</em>

</p>
```

*FILE: reference.html

TEXT ELEMENTS

- **Strong** ``
- Subscript ``
- Superscript ``
- Inserted text `<ins></ins>`
- Deleted text ``
- **Mark** `<mark></mark>`

```
<p>
  <strong>Strong</strong>
  Text w/ <sub>Subscript</sub>
  Text w/ <sup>Superscript</sup>
  <ins>Inserted</ins>
  <del>Deleted</del>
  <mark>Highlighted</mark>
</p>
```

*FILE: reference.html, FCMMAP > index.html

SPECIAL CHARACTERS

- Copyright `©`
- Registered `®`
- Trademark `™`
- Arrows
 - Left `←`
 - Right `→`
 - Up `↑`
 - Down `↓`

```
<p>
  &copy;
  &reg;
  &trade;
  &larr;
  &rarr;
  &uarr;
  &darr;
</p>
```

*FILE: reference.html

OTHER ELEMENTS

- **Horizontal Line** `<hr>`
 - Width `width`
- **Images** ``

Images Cont...

```
<body>
  <br>

  <hr width="500px">
  <p>some text</p>
</body>
```

*FILE: reference.html

REVIEW

- How many header sizes are available?
- Which tag do you use to break a line in a paragraph?
- What are the tags for bold, underline, italicized?
- Bold has the same effect as...
- Italicized has the same effect as...
- Underline has the same effect as...
- You can strike through a text using which tag?
- How do you insert a horizontal line?

IMAGES

- **Images** ``
 - Source `src`
 - Alternate Description `alt`
 - Height `height` (px)
 - Width `width` (px)
- **Filetypes**
 - JPG, GIF, PNG, SVG (?)

```
<body>
  <br>
</body>
```

*FILE: reference.html

MEASUREMENTS

- For width, length, padding, font-size, borders, etc.
- **Relative vs Absolute**
 - Pixels
 - Percentage
 - Em
 - Pt
 - In
 - Cm

```
<body>
  <hr width="50px">
  <hr width="50%">
  <hr width="50em">
</body>
```

*FILE: reference.html

MEDIA

- **Videos**
 - Youtube
 - From file (mp4, mpeg, etc.)
- **Sounds**
 - (mp3, wav, etc.)

* FILE:
reference.html

```
<body>
<video width="320" height="240" controls>
<source src="file.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>

<audio controls>
<source src="file.mp3" type="audio/mpeg">
Your browser does not support the audio tag.
</audio>

</body>
```

LINKS

- **Text Links**
- **Image Links**
- **Relative & Absolute**
- **Targeted** (`_new`)
- **Anchored** (`#name`, `!#`)
- **Email Links**
- **Download**
- **Title**

```
<body>
<a href="http://url.com#anchor"
target="_blank"
title="description">
Text Link</a>

<a href="http://url.com#anchor"
target="_blank"
title="description">
</a>

</body>
```

*FILE: references.html, anchorlinks.html

REVIEW

- Where can you use measurements?
- What are the differences of relative vs absolute measurements? Give examples of each.
- What if you want to reduce or enlarge an image?
- What is an alternate description for?
- What tag should you use to put multimedia in a page?
- What are relative vs absolute links?
- How do you create a text link? How do you create an image link?
- How can you make an email link?
- How do you make a link open in a new window?
- How do you create an anchor link?

LISTS (INTRO)

- **Ordered Lists**
- **Unordered Lists**

```
<body>
<ol>
<li>What</li>
<li>Where</li>
<li>When</li>
</ol>

<ul>
<li>What</li>
<li>Where</li>
<li>When</li>
</ul>

</body>
```

*FILE: references.html

TABLES (INTRO)

- Table `<table>` `</table>`
- Rows `<tr>` `</tr>`
- Columns `<td>` `</td>`
- Table Header `<th>` `</th>`
- Rowspan `rowspan`
- Colspan `colspan`

* FILE:
reference.html

```
<body>
<table>
<tr>
<th>Activity</th>
<th colspan="2">Dates</th>
</tr>
<tr>
<td>Self Defense Seminar</td>
<td>5/12/15</td>
<td>9/08/15</td>
</tr>
</table>

</body>
```

FORMS

- **Form** `<form>` `</form>`
- **Input** `<input>`
- **Textarea** `<textarea>` `</textarea>`
- **Select** `<select>` `</select>`
- **Button** `<button>`

*FILE: references.html,
fcmmmap> contact.html

```
<form
action="MAILTO:someone@example.com"
method="post" enctype="text/plain">

<select>
<optgroup label="Group 1">
<option value="Op1" selected>Option1</a>
<option value="Op2">Option2</a>
</optgroup>
<optgroup label="Group 2">
<option value="Op3">Option3</a>
</optgroup>
</select>

<textarea name="message" rows="10"
cols="30">
Type your message here.
</textarea>
<label>Label</label>
<input type="text" name="fullname">
<button type="button">Click me!</button>

</form>
```

INPUT

- Empty element
- Type
 - button
 - checkbox
 - color
 - date
 - datetime
 - email
 - file
 - number
 - password
 - radio
 - range
 - reset
 - search
 - submit
 - tel
 - text
 - time
 - url

* FILE: references.html

DIVS

- Tool for divisions or selections.
- Useful for applying CSS to a certain group of elements.
- Very useful for creating layouts

```
<body>
  <div class="class">
    <h3>Text</h3>
    <p>Paragraph.</p>
  </div>
</body>
```

* FILE: reference.html, divexercis.html

REVIEW

- What's the difference between an ordered list and an unordered list?
- How do you create a table? How would you sequence <td>, <table> and <tr>?
- What do you use to merge columns? What do you use to merge rows?
- What are div tags? What can they be used for?
- Give the tags for the following:

STYLE GUIDE > LOGICAL & CLOSE ALL TAGS

```
<table>
  <tr>
    <td>
      <p>blah blah blah</p>
    </td>
  </tr>
</table>
```

Choose wisely

<table> <tr> <td> <p> blah blah blah </table> </tr> </td> </p>

VS

<table> <tr> <td> <p> blah blah blah </p> </td> </tr> </table>

code is poetry

STYLE GUIDE > MINIMALISTIC

```
<span class="par1">
  <p>lorem ipsum blah blah</p>
</span>
```

```
<p class="par1">lorem ipsum blah blah</p>
```

STYLE GUIDE > PROPER INDENTATION STYLE GUIDE > AVOID LONG CODE LINES

```
<table><tr><td>One</td><td>Two</td></tr></table>
```

```
<table>
  <tr>
    <td>One</td>
    <td>Two</td>
  </tr>
</table>
```

```
<table>
  <tr>
    <td>
      One
    </td>
    <td>
      Two
    </td>
  </tr>
</table>
```

code is poetry

STYLE GUIDE > USE COMMENTS

```

.site .skip-link {
  background-color: #f1f1f1;
  box-shadow: 0 0 1px 1px rgba(0, 0, 0, 0.2);
  color: #21759b;
  display: block;
  font: bold 14px/normal "Noto Sans", sans-serif;
  left: -9999em;
  outline: none;
  padding: 15px 23px 14px;
  text-decoration: none;
  text-transform: none;
  top: -9999em;
}

/** 7.0 Accessibility **/
/* must have higher specificity than alternative
color schemes inline styles */
.site .skip-link {
  background-color: #f1f1f1;
  box-shadow: 0 0 1px 1px rgba(0, 0, 0, 0.2);
  color: #21759b;
  display: block;
  font: bold 14px/normal "Noto Sans", sans-serif;
  left: -9999em;
  outline: none;
  padding: 15px 23px 14px;
  text-decoration: none;
  text-transform: none;
  top: -9999em;
}
code is poetry

```

STYLE GUIDE > LOWERCASE / QUOTES

elements attributes values class/id filenames

```

```

*use lowercase, dashes, underscores for filenames

BAD WEBSITE EXAMPLE

REVIEW THE CODE:

view-source:<http://www.dokimos.org/ajff/>

*find 5 bad coding practices

DAY 2: CSS

ADDING COLOR AND STYLE TO YOUR SITE

CSS

Cascading Style Sheets

- How HTML elements will be displayed
- HTML shouldn't have formatting
- Changing style attributes (size, color, design)
- Saves a lot of work / time
- .css file extension

TYPES OF CSS STYLING

• Inline

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<span style="text-align: right;">Right aligned</span>
```

• Internal

```
<head>
  <style>

    body { background-color: linen;}

    h1 { color: maroon; margin-left: 40px;}

  </style>
</head>
```

• External

```
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

CSS FOR PRINTING

FOR: if your webpage has a print-friendly version

```
<link rel="stylesheet" type="text/css" href="style.css"
  media="screen" />
```

```
<link rel="stylesheet" type="text/css" href="printstyle.css"
  media="print" />
```

* Remove non-essential, decorative items

TYPES OF CSS STYLING

Exercises:

INLINE: "sel-bord-style.html"

INTERNAL: "sel-bord-style.html"
& "fonts-links-list.html"

EXTERNAL: "display-box-positioning.html"

CASCADING EFFECT

Browser prioritizes styles according to:

1. Inline style (direct)
2. External and internal style sheets (inside <head> section)
3. Browser default

COMPARISON

HTML

```
<HTML>
<HEAD>
  <TITLE> Site Title </TITLE>
</HEAD>
<BODY>
  <P>Paragraph</P>
</BODY>
</HTML>
```

CSS

```
BODY {background:
  url('imgs/greencrosshatch.jpg');}

A:link {text-decoration: underline #000000;}

P {font-family: Arial, serif; color: red;}
```

CSS SYNTAX

```
P { color: red;
  font-family: arial; }
```

```
selector { declaration }
selector { property: value; }
```

Declarations: ends w/ semicolons
enclosed by curly brackets

SELECTORS

• ELEMENT

```
P {font-family: lucida sans; color: red;}
```

• ID (uses ID attribute)

```
#sampleid {color: blue;}
```

• CLASS (uses class attribute)

```
.sampleclass {color: green;}
```

• ELEMENT.CLASS (targets element w/a particular class)

```
p.class2 {color: purple;}
```

• GROUPED (multi-selector w/ same declarations)

```
p, #sampleid, .sampleclass {color: orange;}
```

*FILE: sel-bg-bord-style.html

SELECTORS / FUNCTION

SELECTOR	EXAMPLE	FUNCTION
element,element	div, p	Selects all <div> elements and all <p> elements
element element	div p	Selects all <p> elements inside <div> elements
element>element	div > p	Selects all <p> elements where the parent is a <div> element
[attribute]	[target]	Selects all elements with a target attribute
:hover	a:hover	Selects links on mouse over (pseudo class)

FUNCTION	EXAMPLE	DESCRIPTION
Calc()	calc(100% - 100px)	Allows you to perform calculations to determine CSS property values

CSS PSEUDO CLASSES

A pseudo-class is used to define a special state of an element.

:active	a:active	Selects the active link
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:visited	a:visited	Selects all visited links

REVIEW

Instead of an = sign, CSS uses `__` to divide property & value.
 Instead of < and >, CSS uses `"__"`.
 Identify the parts of the syntax.

```
P { color: red;
  font-family: arial; }
```

What are the 4 types of selectors?
 What are the 3 types of CSS styling? What are their differences?
 What are the priorities of the browser for CSS?

BACKGROUND

WHAT YOU CAN USE:

- background-color
- background-image
- background-repeat
 - no-repeat / repeat-y / repeat-x

WHAT YOU CAN USE:

- background-attachment
 - scroll / fixed
- background-position
 - top, bottom, right, left, center
- background

WHERE YOU CAN USE:

Body, Divs, H1, Paragraphs, Links, Etc.

* FILE: sel-bg-bord-style.html

COLORS

- Hexadecimal
- Starts with # followed by 6 digits (RGB)
- 0-9, A-F

RED #ff0000	GREEN #00ff00	BLUE #0000ff
WHITE #fff	GREY #333	BLACK #000

```
#background-color {
  color: #ff0000;
}
```

LINKS

a:link
a:hover
a:visited
a:active

SECTORS

```
a:link {color: #ff0000;}
a:visited {color: #ff0000;}
a:hover {color: #0000ff;}
a:active {color: #00ff00;}
```

* Styles are limited for a:visited links, due to security issues.

* FILE: fonts-links-list.html

EXERCISE

bg_col_links.html

FONTS & TEXT

SERIF / SANS SERIF

font-family
font-size
font-style

color
text-decoration
text-align
text-transform
line-height
letter-spacing

EM = multiplier to the
current font size

2EM = 2 x the current font size

Declarations

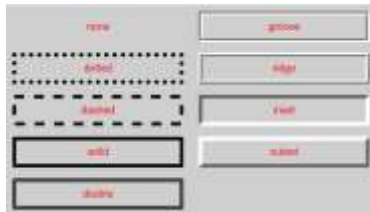
*how to import fonts using google fonts

SHORTHAND: font: font-style font-variant font-weight font-size/line-height font-family;
REQUIRED: Declare font-size first, declare font-family last

*FILE: fonts-links-list.html

BORDERS

- border-top
- border-right
- border-bottom
- border-left
- border
- border-width
- border-style
- border-color



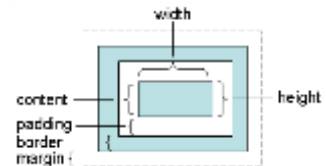
```
.china {
  border-style: solid;
}
```

```
#usa + #mexico {
  border: 1px dashed;
}
```

*FILE: sel-bg-bord-style.html

DIMENSIONS

WIDTH
HEIGHT
MAX-WIDTH
MAX-HEIGHT
MIN-WIDTH
MIN-HEIGHT



MARGIN VS PADDING

All HTML elements are boxes
Both clear areas in or out of the border

MARGIN – area outside border
PADDING – area inside border

Ex:

Width: 150px
Padding: 10 x 2 = 20px
Border: 10 x 2 = 20px
TOTAL = 190px

TOTAL ALLOWABLE FOR MARGIN = 400 - 190 = 210px
210 / 2 = 105px per side margin



*FILE: display-box-positioning.html

MARGIN & PADDING

MARGIN

margin-top: 0px;
margin-bottom: 0px;
margin-right: 0px;
margin-left: 0px;
margin: 0 0 0 0;
margin: 0 0;
margin: 0;

PADDING

padding-top: 0px;
padding-bottom: 0px;
padding-right: 0px;
padding-left: 0px;
padding: 0 0 0 0;
padding: 0 0;
padding: 0;

```
#kim-kardashian {
  padding-bottom: 9999px;
}
```

REVIEW

- Where can you use backgrounds?
- Background-image, background-color, background-repeat, background-attachment can be shortcut into?
- What do the 6 digits in the hexadecimal code represent?
- What is the lowest hex value? What is the highest hex value?
- #00ff00 will give?
- Can you mix border styles? Shortcut for border?
- How many fonts can you use as a value?
- How many google fonts can you import?
- Can you use an image on your links?
- Can you use multiple values for text-decoration?

LISTS

- List-style-type / list-style
 - OL - upper-roman (I, II, III), lower-alpha (a,b,c), decimal-leading-zero, lower-greek
 - UL - circle, square, disc, none
- list-style-image: square, url(file.gif);

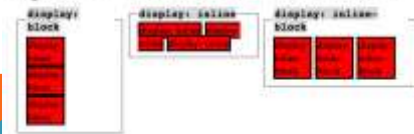
*FILE: fonts-links-list.html

DISPLAY

- Controls how an element is displayed ("type of box")
- Every HTML element is either:
 - Block (<div> <h1> <h6>)
 - Inline (<a>)
- Display: none (invisible but not deleted - no effect on layout)

block vs inline vs inline-block

Below are a bunch of <div style="width: 50px;"> with different display settings.



*FILE: display-box-positioning.html

DISPLAY

- INLINE
 - respect left & right margins and padding, but not top & bottom
 - **cannot have a width and height set**
 - allow other elements to sit to their left and right.
- BLOCK
 - respect all of those
 - force a line break after the block element
- BLOCK-INLINE
 - allow other elements to sit to their left and right
 - respect top & bottom margins and padding
 - respect height and width
 - **similar to inline elements but CAN have a width & height**

```

border: 1px solid black;
display: inline-block;

```

*FILE: display-box-positioning.html

REVIEW

- How to change ul/ol bullets? What are some available bullets for ol? ul?
- Can you use an Image for bullets? How?
- What is the purpose of the display property? What are the four types?
- Which display property shows boxes as stacked?
- Which display property respects height & width?
- Which display property respects margins? Why?

POSITIONING

"How it positions itself in its environment"
Position (static, relative, absolute, fixed)
Specifies type of positioning

```

position: absolute;
visibility: hidden;

```

- Static** - not affected by **top, bottom, right, left** properties; follows normal page flow
- Relative** - "relative to itself", needs positioning otherwise same as static
- Absolute** - removes element from normal flow, allows you to place element anywhere using positioning.



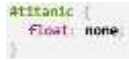
*FILE: display-box-positioning.html

ALIGNING

- Center align (margin: auto)

(always define margin and padding for <body> for the following)

- Left & right align (absolute)
- Left & right align (float)



*FILE: display-box-positioning.html

FLOAT

- Float – wraps elements around the object (box objects)
- Float (left, right)
- Elements reflow
- Clear – elements are told not to flow around the floated element
 - specifies which sides of an element's box other floating elements are not allowed.
- Not applicable to absolute positioned elements

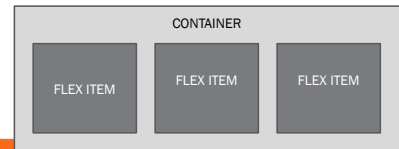


*FILE: display-box-positioning.html

EXERCISE: FLOAT LAYOUT

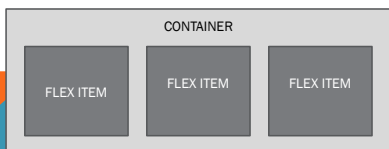
FLEXBOXES

- Allows your elements to be easily laid out for different screen / display sizes
- Gives give the container the ability to alter its child items' width/height (and order)
- Created to replace the use of box model & floats for layout
- APPLIED TO THE CONTAINER OF ELEMENTS TO BE LAID OUT**



FLEXBOXES

- DISPLAY (container):** flex | inline flex
- DIRECTION (body):** rtl (right to left)
- FLEX-DIRECTION(cont.):** row | row-reverse | column | column-reverse
- JUSTIFY-CONTENT(cont.)** – distribution of items (horizontally)
 - flex-start | flex-end | center | space-between | space-around
- ALIGN-ITEMS(cont.)** – vertical distribution
 - flex-start | flex-end | center



EXERCISE: FLEXBOXES LAYOUT

REVIEW

What is the idea of the box model?

What is outside the border? What is inside the border?

Say you create a 100px X 120px box, 5px (padding), 10px (margin), 5px (border).
What is the total height?

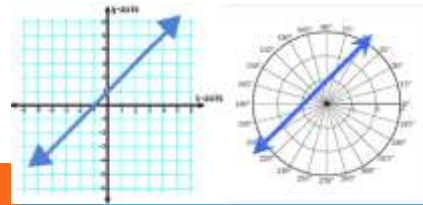
Which position removes an element from the natural flow?

TRANSFORM

ROTATE(_deg)

SCALE(_) [percentage]

SKEW (_deg _deg) | x & y values respectively, horizontal x vertical|

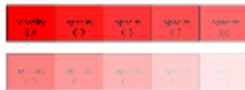


OPACITY

100% => 1.0

50% => 0.5

0% => 0.0



SELECTOR { opacity: (value) }



TEXT-SHADOW

ANIMATIONS (KEYFRAMES)

lets an element gradually change from one style to another

● **keyframes** = gradually changes current style > new style at certain times,
When an animation is finished, it changes back to its original style.

animation-delay property specifies a delay for the start of an animation.

animation-iteration-count specifies times an animation should run.

ANIMATIONS (KEYFRAMES)

1. Element

```
keyframes: 100%;
background-color: red;
background-color: blue;
background-color: green;
background-color: red;
```

2. Animation

```
background-color: red;
background-color: blue;
background-color: green;
background-color: red;
```

ANIMATIONS (KEYFRAMES)

*Using fractions/percentage of the animation duration

1. Animation

```
keyframes: 100%;
0% {background-color: red;}
50% {background-color: blue;}
100% {background-color: green;}
100% {background-color: red;}
```

2. Element

```
keyframes: 100%;
background-color: red;
background-color: blue;
background-color: green;
background-color: red;
```

ANIMATIONS (KEYFRAMES)

*Using fractions/percentage of the animation duration w/ movement

1. Animation

```

@keyframes example {
  0% {background-color: yellow; left: 0px; top: 0px;}
  25% {background-color: yellow; left: 250px; top: 0px;}
  50% {background-color: green; left: 500px; top: 0px;}
  75% {background-color: green; left: 750px; top: 0px;}
  100% {background-color: red; left: 1000px; top: 0px;}

```

2. Element

```

div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

```

CSS3 ANIMATION GENERATOR

<http://css3generator.com/>
<http://www.css3maker.com/css3-animation.html>

CREATING YOUR OWN NAV BAR

1. Create list with proper hierarchy & nesting
2. Set style properties of UL & LI accordingly.
3. Set link properties of menu items (dimensions, hover, etc.)
4. Style submenu
5. Make submenu disappear
6. Make submenu appear on hover

*FILE: menu.html

SEMANTIC HTML5 ELEMENTS

Semantic (descriptive)

- <header> </header>
- <nav> </nav>
- <section> </section>
 - Introduction, content, slider, sports section, etc.
 - Ads section, business section, etc.
- <article> </article>
 - Posts, Article (independent)
- <main> </main>
- <aside> </aside>
- <footer> </footer>

- Why Semantic? Uniformity



*FILE: divexercises.html

INTRO TO RESPONSIVE DESIGN



480px
below

481 to 768 pixels

769 to 1232 pixels

3 BASIC DEVICES

MEDIA QUERIES

- allow you to customize the presentation of your web pages for a specific range of devices
- Logical expression

```

/* Smartphones (portrait and landscape) */
@media screen and (min-width: 320px) and (max-width: 480px){
  /* CSS styles */
}

```

MEDIA QUERIES

MIN-WIDTH

media only screen and (min-width: 330px) {...}

"If [device width] is greater than or equal to [specified #], then do {...}"

So if the actual "device width" is 320px this condition will return false.

"If 320px is greater than or equal to 330px, then do {...}"

MIN-WIDTH is the minimum width at which the styles will be shown.

A screen narrower than the specified number will not use the styles associated with that rule.

MEDIA QUERIES

media only screen and (max-width: 330px) {...} Translates to:

"If [device width] is less than or equal to [specified #], then do {...}"

So based on this second example, if the "device width" is 320px the condition is true:

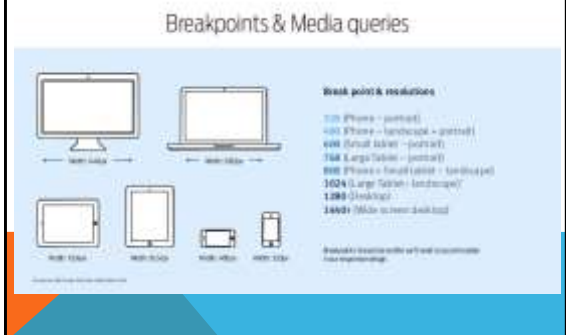
"If 320px is less than or equal to 330px, then do {...}"

MAX-WIDTH is the maximum width at which these styles will be shown. A screen wider than the specified number will not use the styles associated with that rule.

ADD EXTENSIONS

1. Visit the [Chrome Web Store](#).
2. Find and select the extension you want to install.
3. Click **Add to Chrome**.
4. A dialog will appear to show you data that the extension will be able to access.
5. Click **Add** to grant the extension access to your data and install the extension.

BREAKPOINTS & DEVICE WIDTHS



BOOTSTRAP

- Bootstrap is a free front-end framework
- Bootstrap **Includes HTML and CSS** based design templates for typography, forms, buttons, tables, navigation, etc.
- Lets you create responsive designs
- "repurposes some tags"
- **Save lots of time**
- **Responsive features**
- **Consistent design**
- **Easy to use**
- **Compatible with browsers**
- **Open Source**

DOWNLOAD & SET UP

- **BOOTSTRAP** (<http://getbootstrap.com/>)
 - Download minified
 - Unzip, place css & jquery files in directory
 - Link
- **JQUERY** (<https://jquery.com/>)
 - Compressed
 - Javascript of bootstrap uses JQuery

GETTING STARTED

• Establish doctype

• Insert the ff. In <head>:

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

Use viewing device's width

Initial zoom when loaded (100%)

• Import the CSS & JS libraries

1. bootstrap.min.css using <link>
2. bootstrap-theme.min.css using <link>
3. jquery.min.js using <script>
4. bootstrap.min.js using <script>

*meta viewport = viewports & zooming properties

viewport = user's visible area of a web page

TEST USING H1

USING CONTENT DELIVERY NETWORK (CDN)

BENEFITS:

- No hosting required for bootstrap files
- No downloading necessary

CDNs

- hosts Bootstrap's files on multiple servers spread across the globe and when a user requests the file, it will be served from the server nearest to them.

TEXT

```
<h1> - <h6>, <h1> - <h6> + <small> (lighter)
<small>
<mark> - highlight
<blockquote> (left aligned) / .blockquote-reverse (right aligned)
<footer>
<code>
<kbd>
```

TEXT (MEANING THRU COLORS)

```
.text-muted (grey)
.text-primary (blue)
.text-success (green)
.text-info (dk. blue)
.text-warning (gold)
.text-danger (red)
.small (85% of inherited)
.text-left
.text-center
.text-right
.text-justify
```

Background colors

```
.bg-primary
.bg-success
.bg-info
.bg-warning
.bg-danger
```

IMAGES

```
.img-rounded
.img-circle
.img-thumbnail
.img-responsive
```



TABLES

ELEMENT: <table>

ATTRIBUTE: ,table (class)

OPTIONS: ,table-striped, ,table-bordered, ,table-hover

MAKE A RESPONSIVE TABLE:

- Enclose your <table> inside a <div> w/ a "table-responsive" class

LIST GROUPS

PAGINATION

ELEMENT: ``

ATTRIBUTE: `pagination (class)`

PAGER

ELEMENT: ``

ATTRIBUTE: `pager (class)`

ends of page

ELEMENT: ``

ATTRIBUTE: `previous (class), next (class)`

BUTTONS

ELEMENT: `<button>`

ATTRIBUTES: `btn(class)` + `button (type)`

OPTION CLASSES: `btn-success`, `btn-danger`, `btn-warning`, etc.

BUTTON LINK

ELEMENT: `<a>`

ATTRIBUTES: `btn(class)` + `btn-link (class)` + `button (role)`

BUTTONS

Button	Class	Description
	<code>btn btn-default</code>	Default gray button with gradient.
	<code>btn btn-primary</code>	Provides extra visual weight to indicate primary action button in a set of buttons.
	<code>btn btn-info</code>	Can be used as an alternative to the default button.
	<code>btn btn-success</code>	Indicates a successful or positive action.
	<code>btn btn-warning</code>	Indicates caution should be taken with the action.
	<code>btn btn-danger</code>	Indicates a dangerous or potentially negative action.
	<code>btn btn-link</code>	Emphasizes a button by making it look like a link while maintaining button behavior.

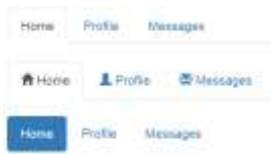
[HTTPS://HACKERTHEMES.COM/BOOTSTRAP-CHEATSHEET/](https://hackerthemes.com/bootstrap-cheatsheet/)

NAV

TABS

ELEMENT: ``

ATTRIBUTE: `NAV` + (`nav-tabs`, `nav-pills`, & more)



NAVBAR

.NAVBAR

- `.NAVBAR-DEFAULT`
- `.NAVBAR-INVERSE`
- `.NAVBAR-LEFT`
- `.NAVBAR-RIGHT`
- `.NAVBAR-FIXED-TOP`



NAVIGATION BAR (PT1)

STANDARD NAVIGATION:

```
<nav class="navbar navbar-default"> </nav>
```

WEBSITE NAME:

```
<div class="navbar-header">
  <a class="navbar-brand" href="#">WebSiteName</a>
</div>
```

MENU ITEMS:

```
<ul class="nav navbar-nav">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">Page 1</a></li>
</ul>
```

NAVIGATION BAR (PT1)

ELEMENT: <nav>

ATTRIBUTE: navbar (class) + navbar-default (class)

OPTION: container-fluid (class), navbar-inverse (class)

NAVBAR HEADER (SITE NAME)

ELEMENT: <div>

ATTRIBUTE: navbar-header (class)

ELEMENT: <p> or <a>

ATTRIBUTE: navbar-brand (class) – for proper padding/margin, etc.

NAVIGATION BAR (PT2)

CREATE YOUR UL MENU

ELEMENT:

ATTRIBUTE: nav (class) + navbar-nav (class)

.nav – displays elements as block & adds padding

.navbar-nav – alignment of the <a>, adds padding

CREATE A DROPDOWN

ELEMENT:

ATTRIBUTE: dropdown (class)

→ Adds relative position

target the link inside

ELEMENT: <a>

ATTRIBUTE: dropdown-toggle (class), dropdown (data-toggle)

function

ELEMENT: (submenu)

ATTRIBUTE: dropdown-menu (class)

absolute

*span w/ class of caret for arrow

*add class of active in for selected

NAVBAR (PT 3)

ELEMENT: (w/ & <a>)

ATTRIBUTES: nav (class), navbar-nav (class), navbar-right

COLLAPSIBLE NAVBAR

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
        Bars
      </button>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        Menu Items
      </ul>
    </div>
  </div>
</nav>
```

GLYPHICONS

ELEMENT:

ATTRIBUTE: glyphicon (class) + [glyphicon-home (class)]

OPTIONS:

- .glyphicon-envelope (class)
- .glyphicon-print (class)
- glyphicon-user (class)
- glyphicon-ok (class)

See full list @ www.getbootstrap.com website

FORMS

- ELEMENT: `<form>`
- ATTRIBUTE: `form (role)`
- ELEMENT: `<div>`
- ATTRIBUTE: `formgroup (class)` - adds bottom margin
- EMAIL LOGIN**
 - Element: `<input>`
 - ATTRIBUTES: `email (type)`, `form-control (type)`, `email (ID)`, `password (type)`, `pwd(ID)`
 - Element: `<button>`
 - ATTRIBUTES: `submit (type)`, `btn (class)`, `btn-default(class)`
 - *form control - adds style, bg, margin, padding, 100% width for form elements.

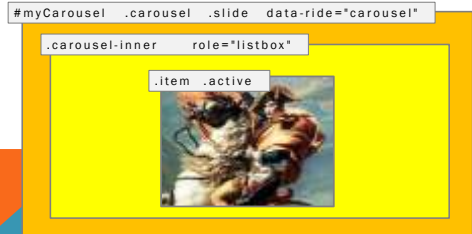
CAROUSEL (SLIDES)

1. Indicators

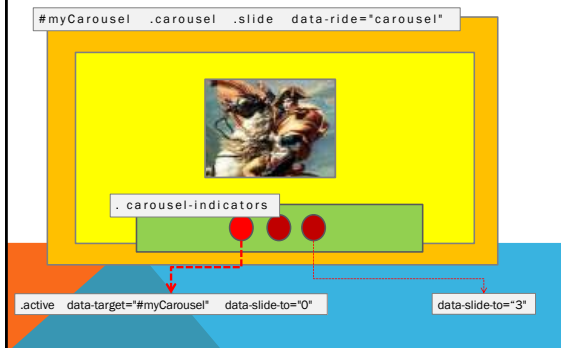
2. Slides

3. Controls

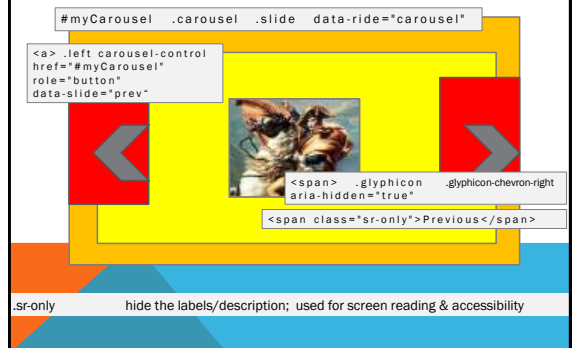
data-ride="carousel" > attribute activates the carousel



CAROUSEL (INDICATORS)

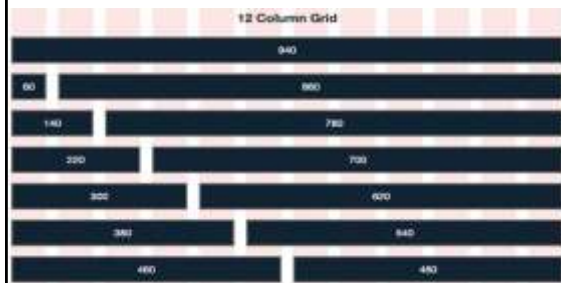


CAROUSEL (CONTROLS)



GRID SYSTEM: 960 GRID

- 12 column layout (streamline responsive layouts)
- based on 960px grid
- 60px columns w/ 10px margin



CONTAINERS & 12 COLUMN GRID

4 Grid Classes

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)

HOW GRID COLUMNS ARE CREATED

Syntax: `.col-(grid class)-number`

EX: `.col-md-4`

Where:

- x is any of xs, sm, md or lg
- y is any of 1, 4, 6, 12

total of y should be 12 always

CONTAINERS & 12 COLUMN GRID

- Works like HTML tables
- Establish rows <div> with **.container** or **.container-fluid** first.



BOOTSTRAP GRID

12 / no. of boxes you need = grid number



CONTAINERS & 12 COL GRID

Feature	Extra small devices (Phones < 768px)	Small devices (Tablets > 768px)	Medium devices (Desktops > 992px)	Large devices (Desktops > 1200px)
Bootstrap 4 Grid system	Phone (<768px)	Small (>768px)	Medium (>992px)	Large (>1200px)
Max container width	None (auto)	750px	970px	1170px
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
Max column width	Auto	~62px	~81px	~97px
Gutter width	15px on each side of a column (i.e. 30px)			

12 COLUMN GRID

CLASSES

- **CONTAINER**
 - constraints on responsive widths, adds 15px padding
- **ROW**
 - provides the columns a place to live
 - acts as a wrapper, must be inside container
 - Negative 15px margin
- **COLUMNS**
 - 15px padding
 - 30px gutter bet. cols



HIDDEN (CLASSES)

hidden-sm
hidden-md
hidden-lg

SUBLIME EXTENSIONS

<https://packagecontrol.io/>
Go to www.packagecontrol.io
Install package control using python code
In sublime go to packages

Wordpress
Bootstrap

RECOMMENDED SITES

- <http://dynamicdrive.com>
- <http://subtlepatterns.com/>
- <http://www.colorcombos.com/>
- <http://sxc.hu/>
- <http://deathtothestockphoto.com/>
- <http://www.freepik.com/>
- <http://css3generator.com/>
- <http://cssmenumaker.com/>
- <http://960gs.com>
- <http://csslayoutgenerator.com/>
- <http://markgreenall.github.io/StringBean/>
- <http://cocycles.com/>
- <http://www.initializr.com/>
- <http://flexboxfroggy.com/>
- styleguides.io/
- <https://thegrid.io/>

ADD

http://www.tutorialspoint.com/bootstrap/bootstrap_navbar.htm
http://www.w3schools.com/bootstrap/bootstrap_carousel.asp
<http://www.webcredible.com/blog/ten-css-tricks-you-may-not-know/>
http://www.sitepoint.com/using-modern-css-to-build-a-responsive-image-grid/?utm_content=buffer8828d&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer
<http://www.sitepoint.com/elight-html5-tags-you-might-not-know/>
http://www.sitepoint.com/a-basics-html5-template/?utm_content=buffer6bc2b&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer
http://www.sitepoint.com/20-docs-guides-front-end-developers-7/?utm_content=buffer5512c&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer
<http://julianmelissas.com/no-gutter-column-trick-for-bootstrap/>
http://www.sitepoint.com/20-docs-guides-front-end-developers-7/?utm_content=buffer5512c&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer