

Intermediate HTML and CSS

part 4

By Lea Ann Bradford for Girl Develop It San Diego

Welcome

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

Some “rules”

- We are here for you
- Every question is important
- Help each other
- Have fun

Introductions

- Lea Ann Bradford
- Software Developer @ Notch8
- Former student of GDI front end series
- Website: leaannbradford.com
- Twitter: @LeaABradford
- LinkedIn: [linkedin.com/in/leaannbradford](https://www.linkedin.com/in/leaannbradford)



Tools We Will Use Today

Web Browser:

- Chrome - preferred
- Safari
- Firefox

Text Editor

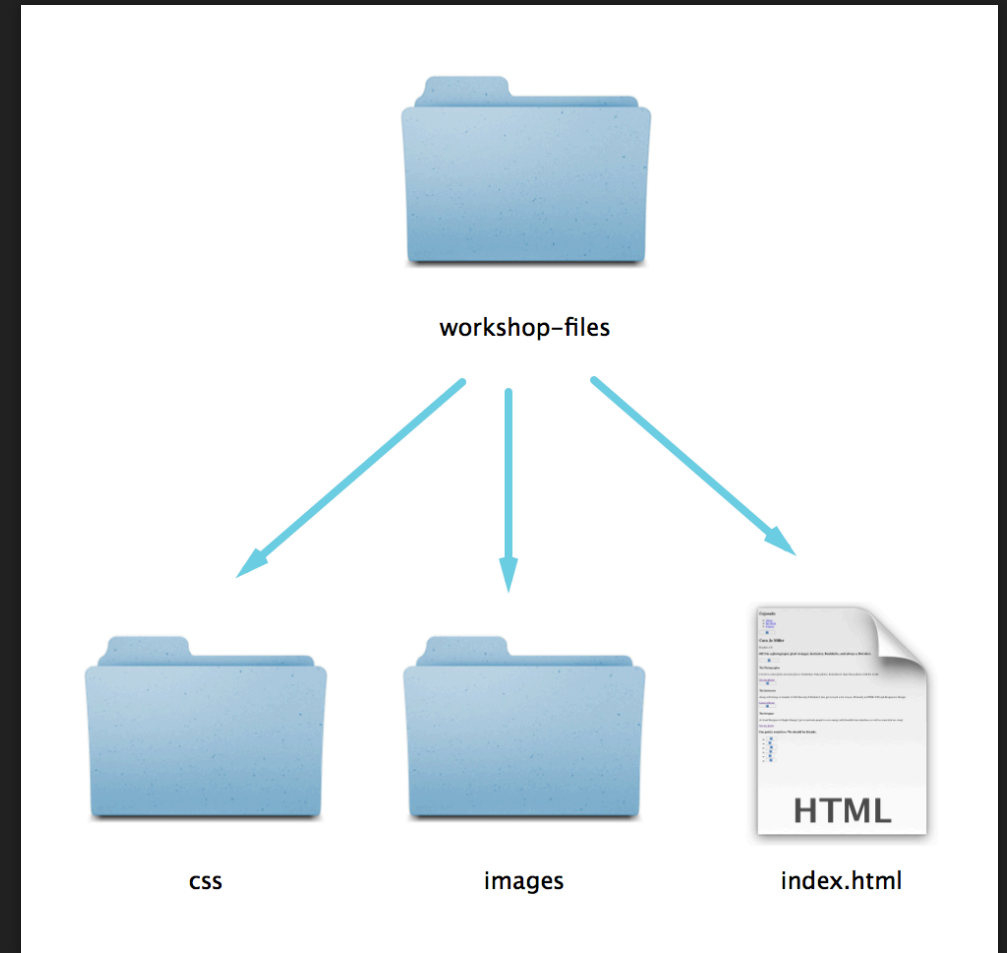
- Sublime Text - preferred
- Atom
- Notepad++

Starter Files

- Download at www.leanannbradford.com
- Includes starter HTML and CSS files
- We also included some image files and code snippets to help you

Folder Structure

- Our folder structure is set up in a commonly used format



Responsive Web Design (RWD)

- RWD is a design approach that suggests that the design & development of a site should respond to the user's behavior and environment.

Responsive Web Design (RWD)

- 91% of U.S. Citizens have their smart phone within reach 24/7
- Smartphone users check their phones 150x every day
- 25% of mobile users in the U.S. don't even own a laptop or desktop

Responsive Web Design (RWD)

- RWD modifies the presentation of a site, without modifying the content of the page.
- So no matter what, every user has access to the same information.

Responsive Web Design (RWD)

The Ingredients of RWD

- A Flexible foundation
- Fluid grids
- Flexible images
- CSS Media Queries to make the magic happen

Responsive Web Design (RWD)

Fluid grids

- With fixed-width sites, we have to adjust the height and width of elements manually.
- With fluid grids, the height and width of elements is dependent upon the device resolution.

Responsive Web Design (RWD)

Fluid grids

- First, we define a maximum width for the container
- Then, we divide the content up into a set of columns, usually 12
- Then, we design elements with proportional widths and heights instead of being stuck with specific pixel dimensions
- Whenever the device width changes, the grids change in width to scale with the device

Responsive Web Design (RWD)

Flexible Images

- Text scales easily on smaller devices, but images are a bit tricky.
- Images will overflow their container elements if they're too big for them.
- That's annoying.

Responsive Web Design (RWD)

Flexible Images

- By using max-width on images, they will only expand to the size of their parent.
- If their parent has no width, it will just expand to the width of the viewport.

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

Responsive Web Design (RWD)

Media queries

- Media queries apply certain CSS in certain situations.
- Print Media
- iPhone
- iPad
- Even larger screens

Responsive Web Design (RWD)

Media queries

- Include media queries last
- They will overwrite previous styles because they are last in the cascade.

Responsive Web Design (RWD)

Media queries

- Rather than looking for a type of device, they look at the capability of the device. You can use them to check for all sorts of things.
- Width & height of the viewport
- Width & height of the device
- Orientation - landscape or portrait
- Resolution - retina or normal

Responsive Web Design (RWD)

Media queries

- Let's take a look at what one looks like
- For screen sizes 901 pixels wide and lower, we are adjusting the size of our h2 in the hero section

```
@media only screen and (max-width: 901px) {  
  #hero h2 {  
    font-size: 200%;  
  }  
}
```

Responsive Web Design (RWD)

Media queries

- Let's make some adjustments to your headers

```
@media only screen and (max-width: 901px){  
  #hero h2 {  
    font-size: 200%;  
  }  
  #main-content h3 {  
    font-size: 150%;  
  }  
}
```

Responsive Web Design (RWD)

Media queries

- Let's collapse down our three column layout for smaller screens

```
@media only screen and (max-width: 768px) {  
    .flex-container {  
        display: block;  
    }  
    .column {  
        width: 100%;  
        padding: 0;  
        margin: 0 0 40px 0;  
    }  
}
```

Responsive Web Design (RWD)

Media queries

- Let's adjust the width of our nav inside the same media query
- We will also need to adjust the size of our padding on the body

```
@media only screen and (max-width: 768px) {  
    nav {  
        width: 25%;  
    }  
    body {  
        padding-left: 25%;  
    }  
}
```

Responsive Web Design (RWD)

Media queries

- Finally, let's make our nav move to the top of the screen for small devices like phones
- Let's add a border to the bottom of our nav items
- We need to remove the padding from the body

```
@media only screen and (max-width: 500px) {  
  nav {  
    position: relative;  
    width: 100%;  
    text-align: center;  
  }  
  nav li {  
    border-bottom: 1px solid #e6ac00;  
  }  
  body {  
    padding: 0 ;  
  }  
}
```

Let's Develop It

- Let's make your site responsive
- Collapse your site and see if there are any other break points where you would like to change your design
- Add a media query and make style changes

Questions?