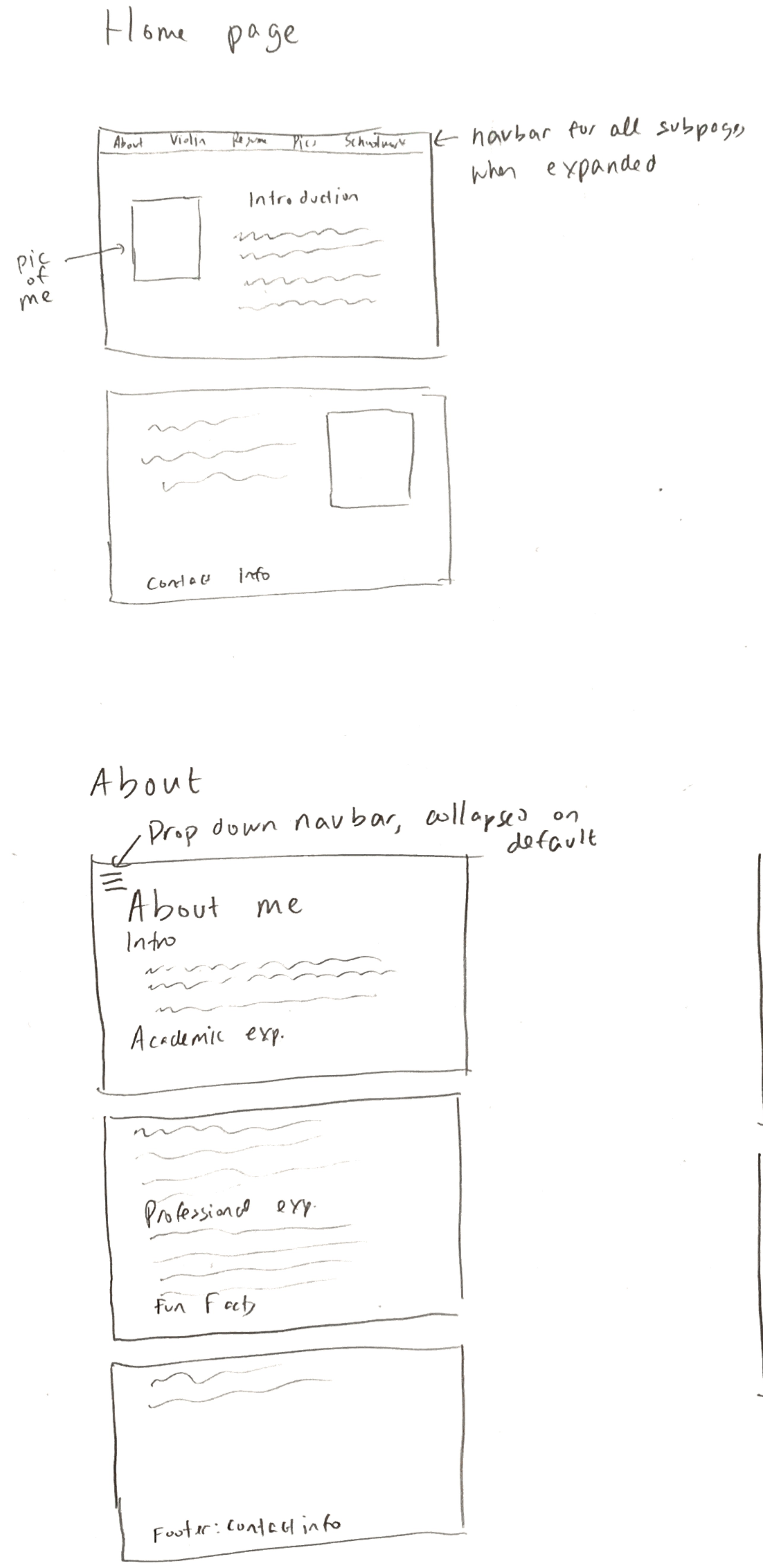
Fundamentals of Web-Based Multimedia Communications:

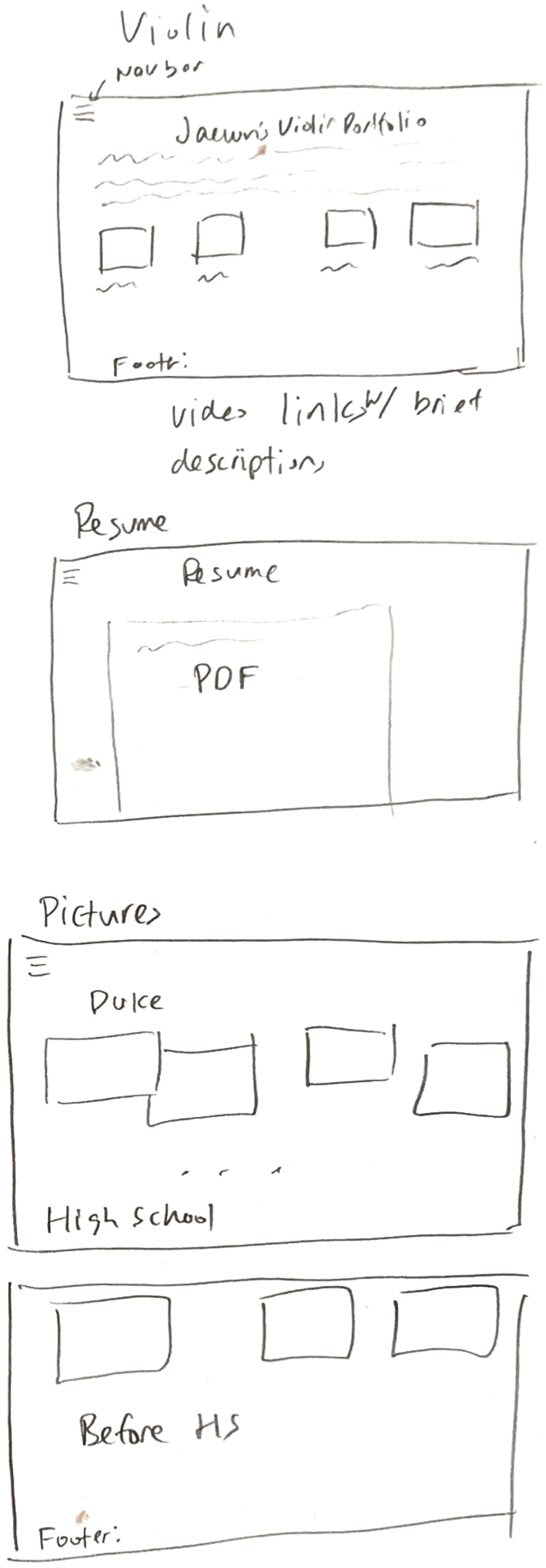
Imagining the Internet's Interface

Wireframes, Site Maps, Responsive Designs

Wednesday 9/23, 1:45-3:00pm
Smith Warehouse Bay 12 A228
CMAC 240L, CMAC 740L, ISS 240L, ISS740L, VMS288L, VMS788L

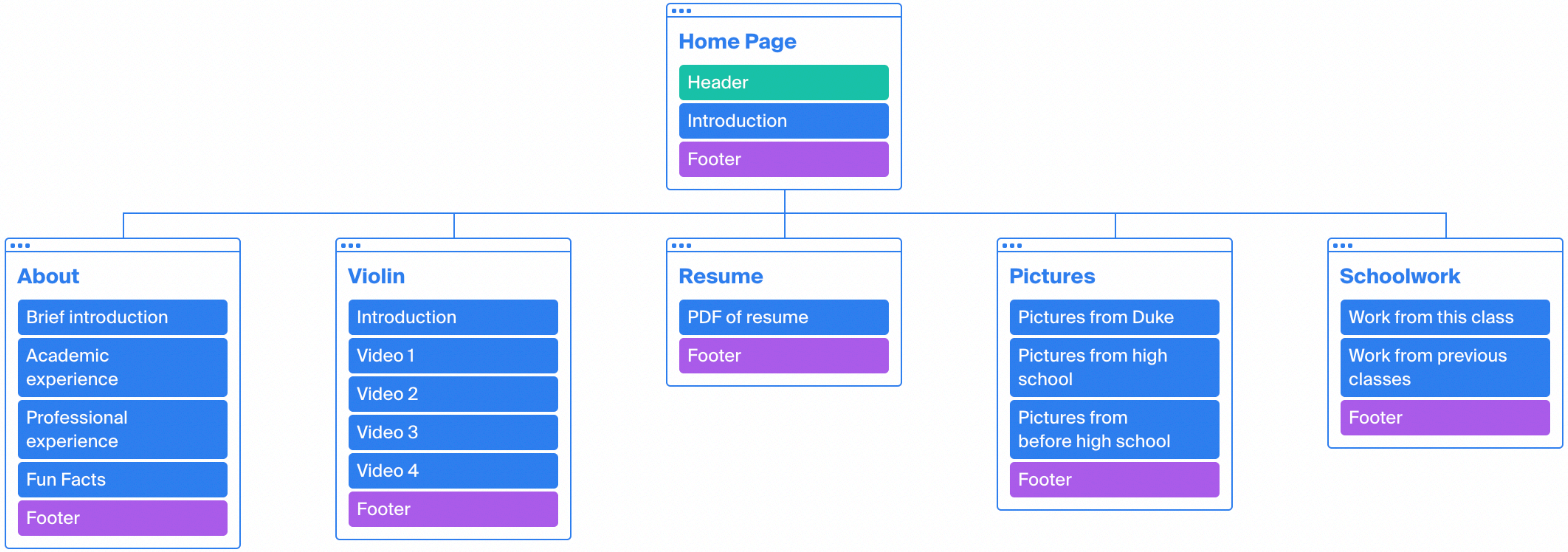
Wireframes and Site Maps





Wireframe

Image credit: Jaewon Jung



Home Page
- Header
- Introduction
- Footer

About
- Brief introduction
- Academic experience
- Professional experience
- Fun Facts
- Footer

Violin
- Introduction
- Video 1
- Video 2
- Video 3
- Video 4
- Footer

Resume
- PDF of resume
- Footer

Pictures
- Pictures from Duke
- Pictures from high school
- Pictures from before high school
- Footer

Schoolwork
- Work from this class
- Work from previous classes
- Footer

Site Map

Image credit: Jaewon Jung

Some review:

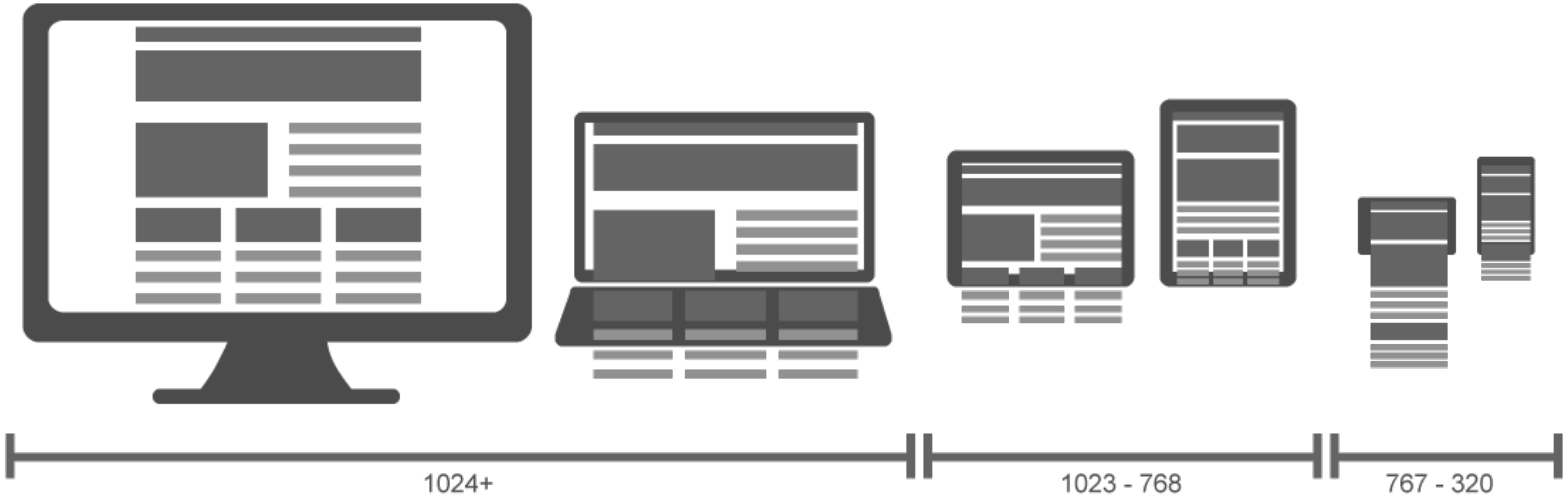
What is it like to build a website?

- It feels like **Freedom!** *Tears of Joy*
- It feels like **Frustration** *Just Tears* (there are so many ways to do a thing and yet I can't do the thing that I want!)
- It feels **Collaborative…** a bunch of other code supports your code
- It is **Always changing!**
- **Must *learn how to learn:*** what is the syntax again? How do I use this code I found off another site? What happens if play with these property values?

We now know how to:

- Create a **File Structure** for a website
- Create **Metadata** for a website in the <head> element
- Create the **Content** for a website using:
    - <body> element
    - <header> element
    - <main> element
    - <section> element
    - <p>, <img>, <blockquote>.. and more!
- **Style** the elements using CSS and:
    - Element selectors
    - Id selectors
    - Class selectors
    - The Universal selector (*) that selects all elements
- Create **File Pathways** to other pages and images
- Use the <alt> attribute to describe content as well as challenge yourself to always use semantic html when necessary (use only divs for styling) to make things **Accessible**
- **Think Creatively** about web design and **Trouble-shoot**

Responsive Web Design



Responsive Web Design

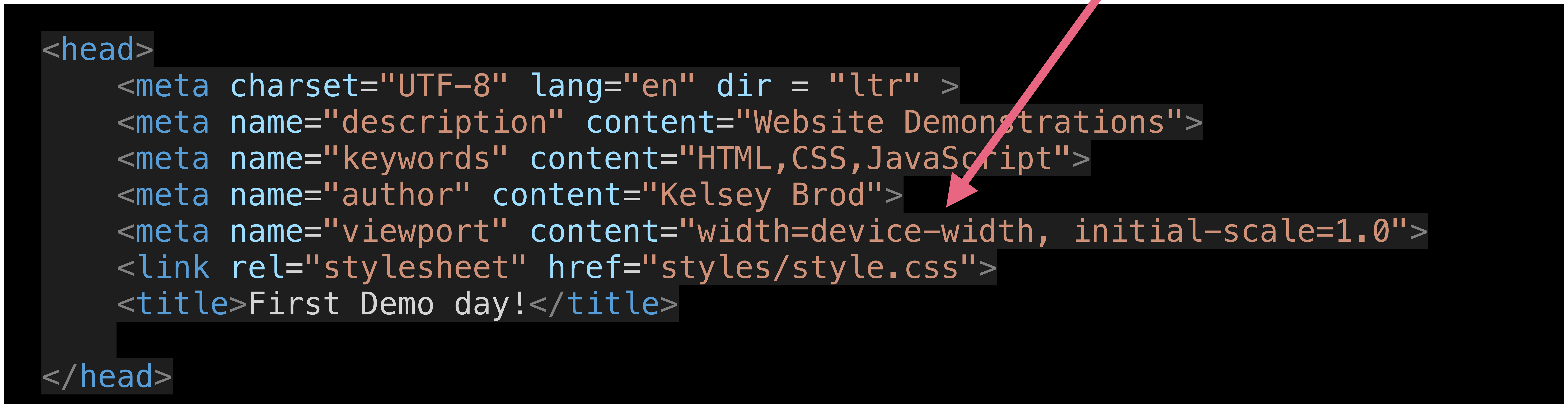
What is it?

- Fluid - uses percentage width for relative sizing (Website Gallery example)
- Adaptive and Responsive - uses @media queries

You will likely use both Fluid/percentage width and @media queries!

Set the Viewport inside <head>

Add as a Meta tag



```
<head>
    <meta charset="UTF-8" lang="en" dir = "ltr" >
    <meta name="description" content="Website Demonstrations">
    <meta name="keywords" content="HTML,CSS,JavaScript">
    <meta name="author" content="Kelsey Brod">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles/style.css">
    <title>First Demo day!</title>

</head>
```

Media Queries

The `@media` rule is used in media queries to apply different styles for different media types/devices.

Media queries can be used to check many things, such as:

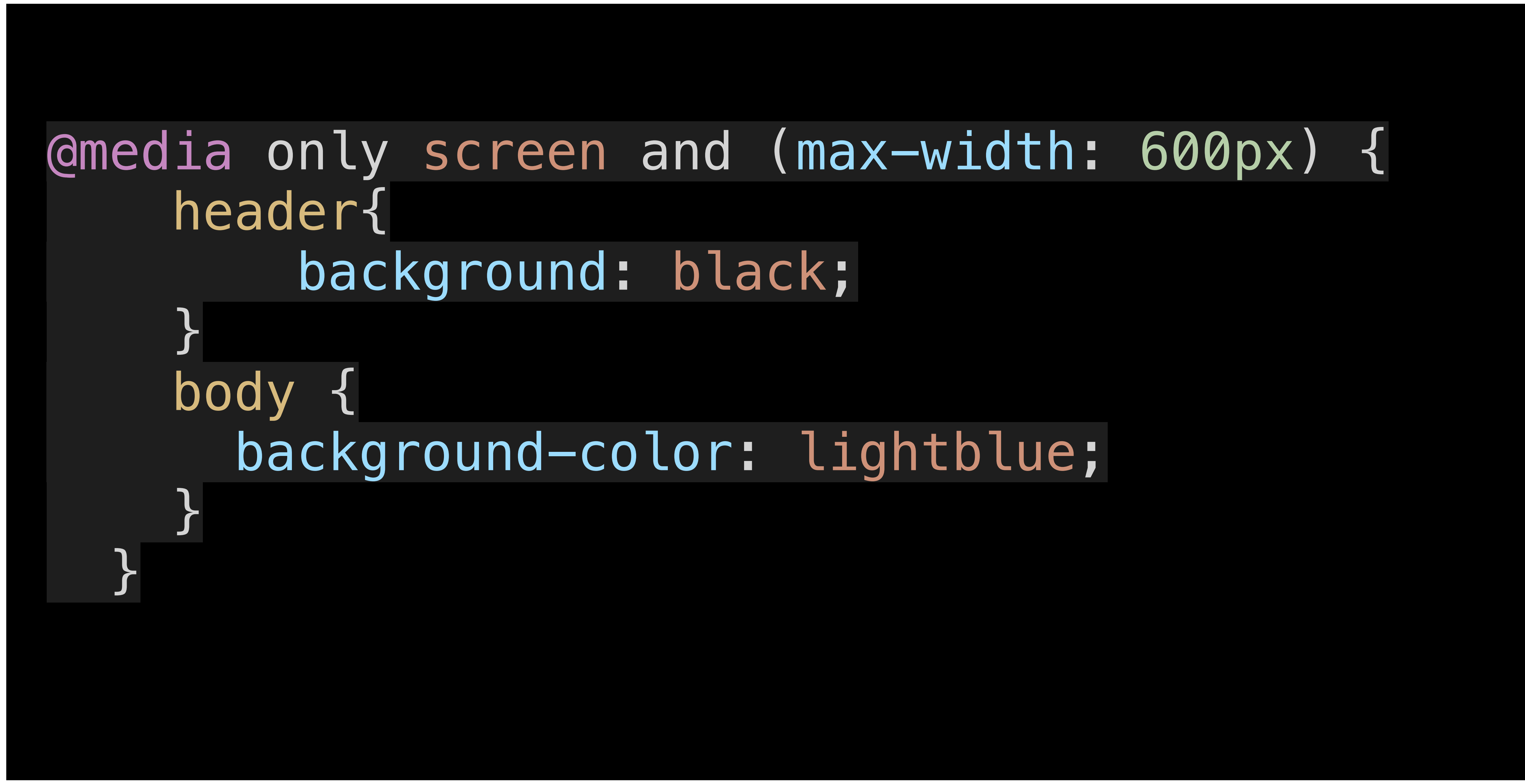
- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries are a popular technique for delivering a tailored style sheet (responsive web design) to desktops, laptops, tablets, and mobile phones.

You can also use media queries to specify that certain styles are only for printed documents or for screen readers (mediatype: print, screen, or speech).

In addition to media types, there are also media features. Media features provide more specific details to media queries, by allowing to test for a specific feature of the user agent or display device. For example, you can apply styles to only those screens that are greater, or smaller, than a certain width.

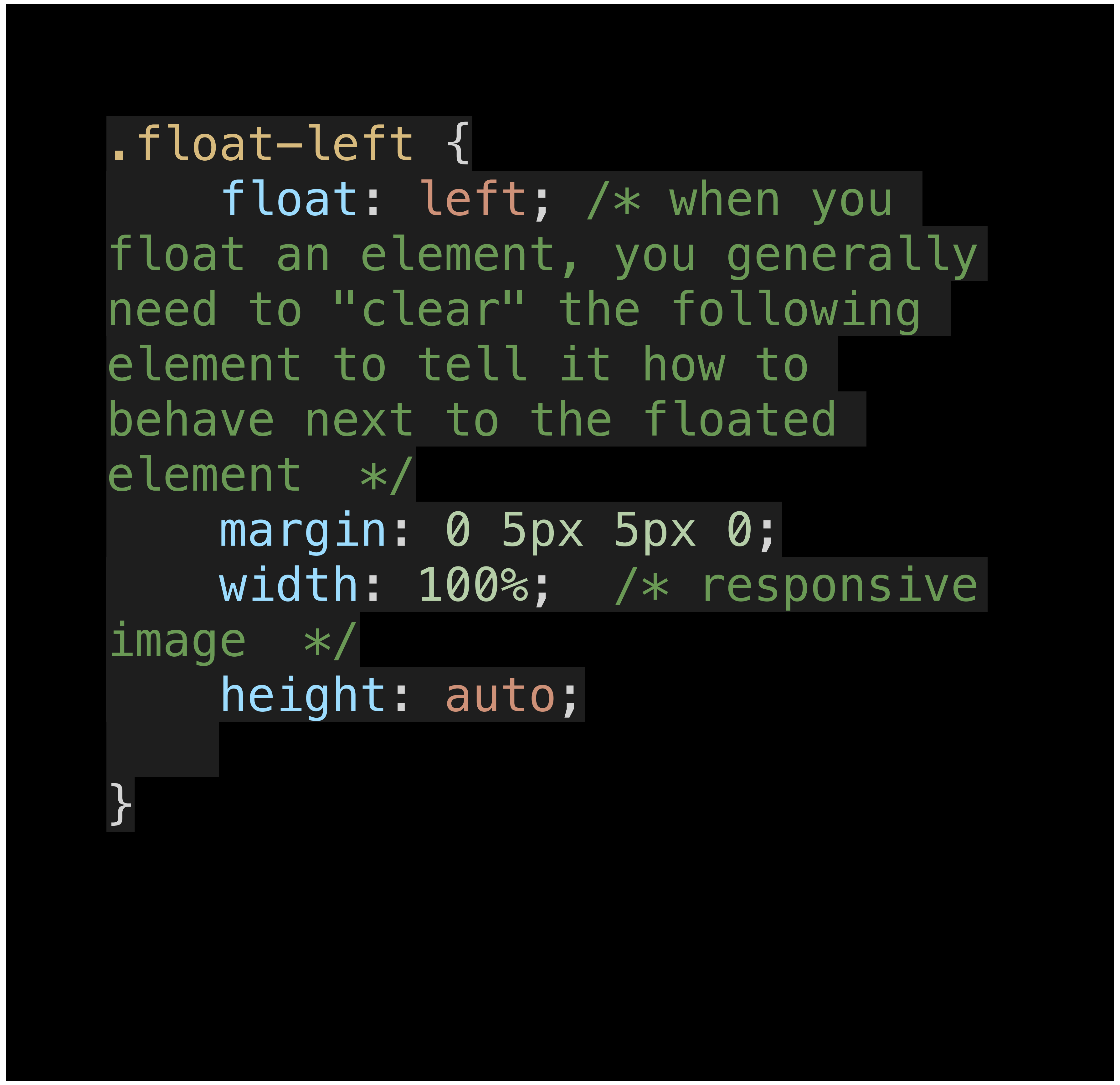
-w3Schools



```
@media only screen and (max-width: 600px) {
    header{
        background: black;
    }
    body {
      background-color: lightblue;
    }
  }
```

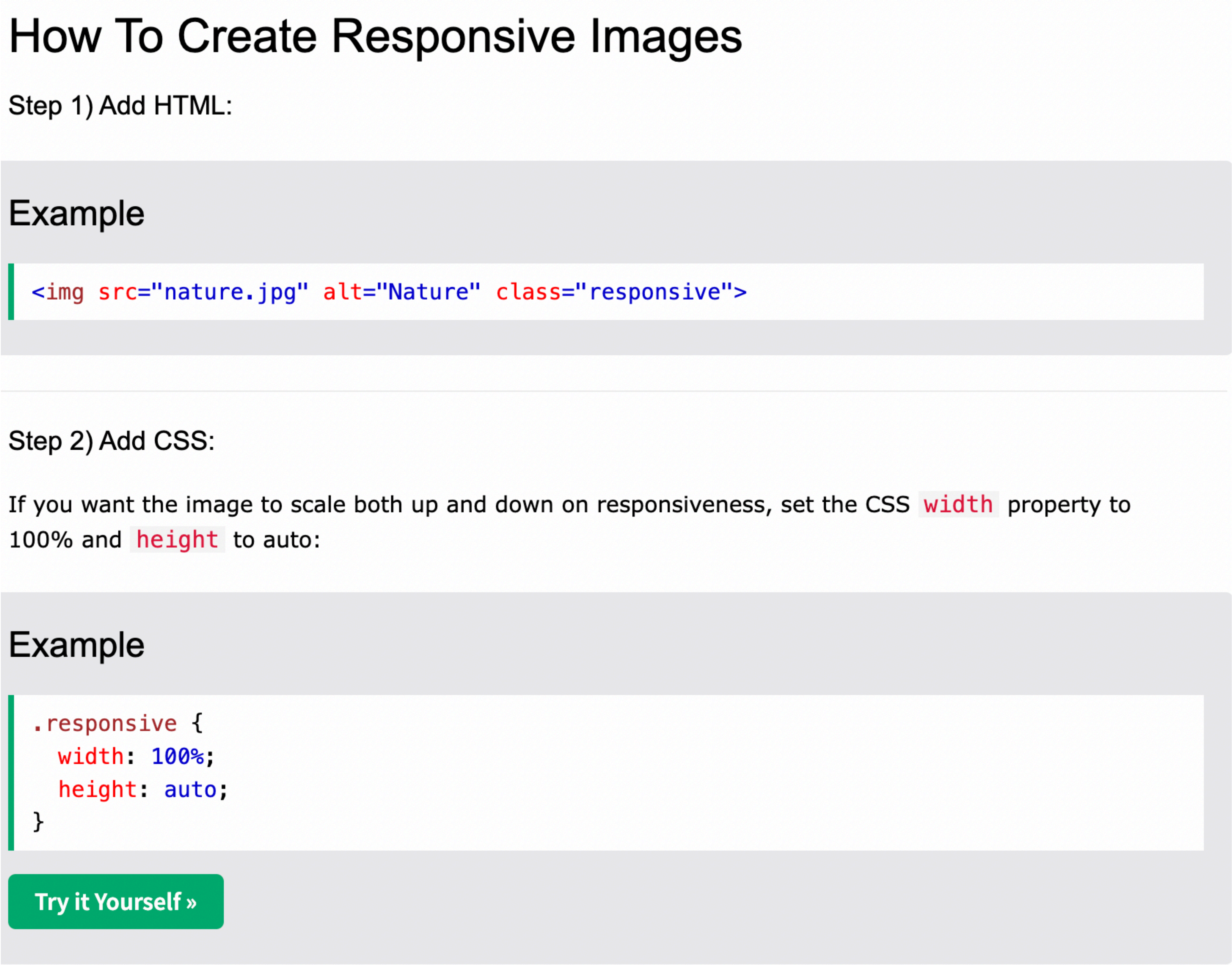
Go to Resources on Sakai to see how the Website Gallery exercise can become responsive

Easy Responsive Images



```
.float-left {
    float: left; /* when you
float an element, you generally
need to "clear" the following
element to tell it how to
behave next to the floated
element  */
    margin: 0 5px 5px 0;
    width: 100%;  /* responsive
image  */
    height: auto;

}
```



How To Create Responsive Images

Step 1) Add HTML:

Example

```
<img src="nature.jpg" alt="Nature" class="responsive">
```

Step 2) Add CSS:

If you want the image to scale both up and down on responsiveness, set the CSS `width` property to 100% and `height` to auto:

Example

```
.responsive {
  width: 100%;
  height: auto;
}
```

-w3Schools

To do: Finish Early Internet Art Project by next class, 9/28
Watch videos on Flexbox and Grid to prepare for the next project