

Web Site Design and Development

Lecture 13

CS 0134
Fall 2018
Tues and Thurs
1:00 – 2:15PM

Normalize.css

- **Normalize.css is a useful project to incorporate into your web pages**
- **In their own words, “Normalize.css makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.”**
- **Adding the normalize.css file as the first stylesheet in a web page will make it easier to write CSS that works the same across all modern browsers.**

How to use Normalize.css

- Go to <https://necolas.github.io/normalize.css/>
- Right click on “Download v8.0.0” and choose “Save link as...”
- Save the file alongside your style.css file for your webpage
- Before any other link elements in your web page’s head element, place a link element for rel=“stylesheet” pointing to normalize.css.

Responsive web design

- **Responsive web design is designing your website so that it works with or “responds to” the size of the screen your user is using.**
- **The three main tools used for are**
 - Fluid layouts
 - Media queries
 - Scalable images

Fluid layouts

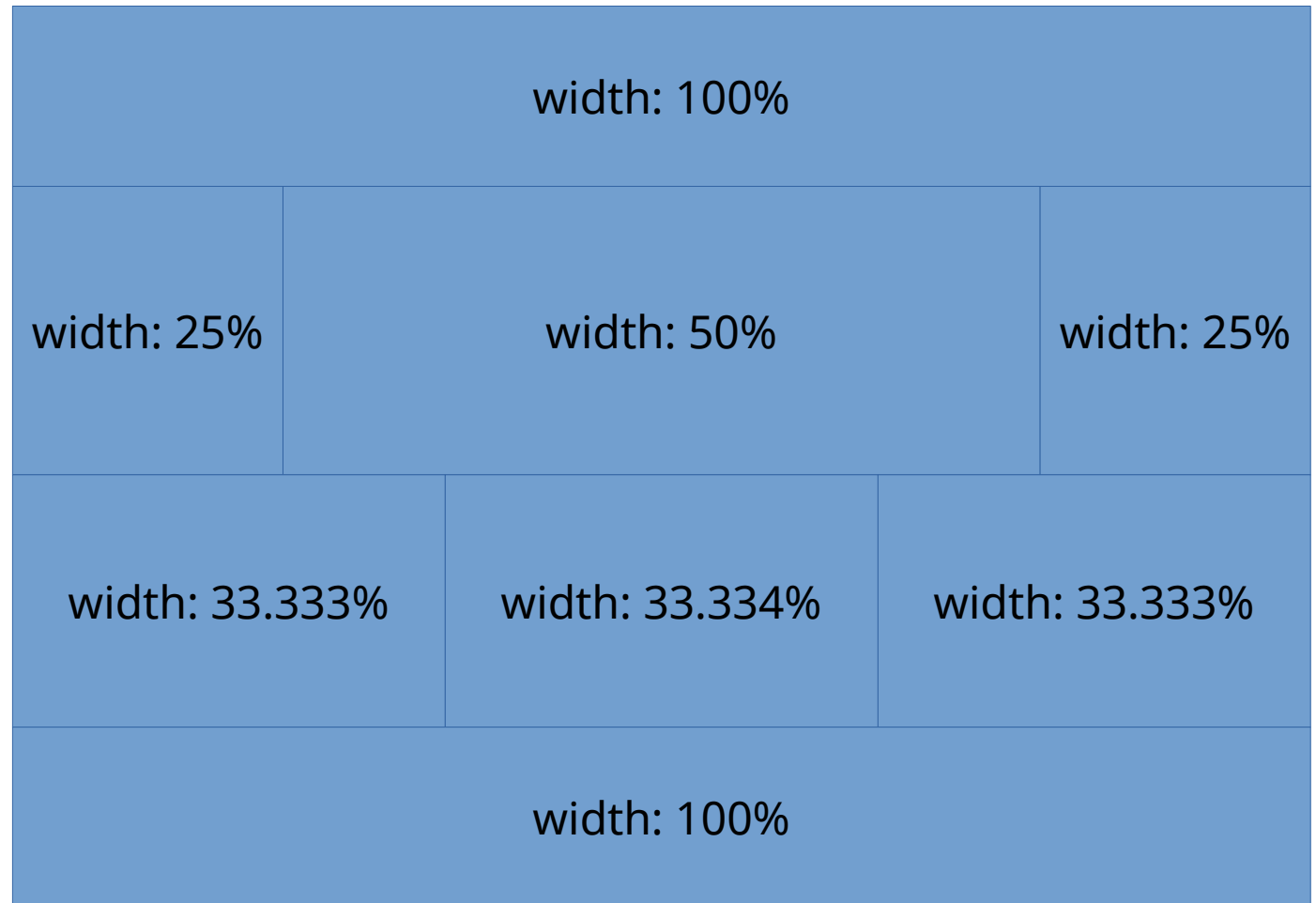
- **There are two types of layouts for web pages**
 - Fixed layout: layout whose elements do not adjust to the user's screen size.
 - Fluid layout: A layout whose elements adjusts to the size of the user's screen size.
- * **A note on screen size, here screen size refers to the size of the browser window. This may not match the size of the monitor or phone's screen.**

Fluid layouts continued

- **A fluid layout is created when you use relative measurements, most commonly percentages, to set the width of the page and its main elements.**
- **It is common to start by setting the page width to a value between 90% and 100%**
- **From here, you will set the remaining element's widths to a percentage of their parent element's width.**

Example fluid layout

```
body {  
  width: 90%;  
}
```



Converting from a fixed layout to a fluid layout

- 1) **Change the page width to a percent between 90 and 100**
- 2) **For every element within the page that has an absolute width set, do:**
 - 1) Note the absolute width of the element as you will need that for the element's children
 - 2) Calculate the percentage width with the equation:
$$(\text{width of element}) \div (\text{width of parent element}) \times 100$$
 - 3) Set width equal to the newly calculated percentage.
 - 4) If an element has left and/or right margins or padding, convert those to percentages as well

Considerations while converting

- Do not round your percentages when possible
- When you set the width of the page, it is a good idea to set a max-width in pixels. This is because fluid layouts start to look odd when they become too wide.
- Borders cannot be specified as percentages so if you have left and right borders on your elements, you will need to adjust your percentages to accommodate.
- An alternative to border that can be used is outline. The outline property is used like border but is drawn outside the element's border and does not affect the total width of an element.



Questions?

Viewport

- **The viewport is the visible section of a web page**
- **In a desktop browser, the viewport is the size of the browser window and will behave as expected**
- **On mobile browsers, the viewport can be larger or smaller than the website trying to be displayed. This is why you will see websites that are shown as tiny versions of themselves. The mobile browser is trying to fit the entire width of the website in the viewport.**

Viewport continued

- **With HTML5, the developer can now control the viewport through a meta element in the web page's head element.**
- **The name for the viewport meta element is "viewport"**
- **The content can consist of 6 properties and their values with each pair of property=value separated by a comma**

Meta viewport content properties

- **width:** The width of the viewport in pixels or the keyword “device-width”. The keyword device-width sets the width to the CSS pixel width of the device at 100% scale
- **height:** The height of the viewport in pixels or the keyword “device-height”. The keyword device-height sets the height to the CSS pixel height of the device at 100% scale
- **initial-scale:** number representing the initial zoom factor of the web page
- **minimum-scale:** minimum zoom factor
- **maximum-scale:** maximum zoom factor
- **user-scalable:** yes to allow user to scale the webpage, no to disallow user to scale th webpage

Meta viewport content properties continued

- **I recommend that you do not use `maximum-scale` and `user-scalable=no`.**
- **This is because there are users who may have trouble viewing your website at your intended size and will need to zoom and may need to zoom further than you intended.**

Meta viewport example

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

CSS pixels?

- **CSS pixels are a logic unit that is used by CSS when you use px in your style rules.**
- **For the purposes of web development, CSS pixels allow us to build websites on devices like phones that have very small physical pixels that would make websites small and hard to use if, for example, a 15px margin was actually 15 tiny physical pixels.**

Other relative measurements for responsive web design

- **In addition to percentages, we have 4 new relative units of measure. These are all based on the viewport**
 - **vh**: 1/100th of the viewport height
 - **vw**: 1/100th of the viewport width
 - **vmin**: 1/100th of the smaller of the height or width
 - **vmax**: 1/100th of the larger of the height or width
- **IE9+ and Edge do not support vmax and us vm for the vmin unit.**



Questions?

Font sizes

- **For responsive design, you want to use ems, percentages, vh, vw, vmin or vmax for your font size**
- **Due to issues in IE9 and earlier, you should use percents for your base font size**
- **Examples**
 - `body { font-size: 100%; }`
 - `h1 { font-size: 2em; }`

Scalable images

- In responsive design, you should use relative measurements for the width of images have them flow with the rest of the web site
- This can keep the images from becoming too large or too small, both of which can affect the usefulness of your website
- Since images like pngs, jpegs and gifs have a fixed size, you may want to limit how large an image can grow by setting the max-width of the image to the width of the image.

Image example



- Feeding an elephant at the Thai Elephant Conservation Center in Lampang Thailand
- This picture has a width of 480px

- The css for this image, if I want it to be full width at the top of my article but not scale too large and become blurry would be:

```
img {  
    width: 100%;  
    max-width: 480px;  
}
```



Questions?



Let's try!