Useful heads-up

- ***Responsive web design*** (RWD) refers to web design that works well on a variety of devices and screen sizes.

- The impetus for doing this is the huge increase in the variety of devices (from smart phone to desktop) and the huge popularity of some of the smaller devices.

    - **flexible grids** – how to base your design on a grid-like arrangement, where the pieces scale for your screen size. We'll use standard templates.
    - **flexible or adaptive images** – how to make your images blow up/shrink appropriately for your screen size
    - **media queries** – how to use different style sheets for different types of media
    - **dynamic content** – content which is responsive to what your user is doing/using; this includes the user rotating a phone/tablet. Some folks don't think of this as "a thing" in RWD – everyone agrees on the first three items.

- the ***viewport*** refers to the screen window.
  RWD inevitably begins with

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

  This sets the size of the viewport (the screen real estate you will be working with) to be that of the device, and sets an initial level of how far you have zoomed in.
  This is not a w3c standard, but it is a de-facto one.
  Eventually you will want to read the details at https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag

- The default font-size for all browsers is 16 points. All of them.
  This is another de-facto standard that is not a w3c standard.

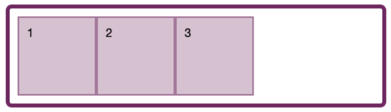
- **CSS pixel** refers to the pixel size you may specify in a style sheet --- and not the actual, physical pixels on a device (which depend on such things as pixel density.)
  If you wish to, you can learn more about this at Peter-Paul Koch's article "A pixel is not a pixel is not a pixel".
  http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html

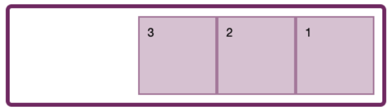
Read:

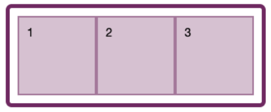
- To see what RWD can do watch the introduction to http://www.lynda.com/CSS-tutorials/About-exercise-files/110716/114016-4.html

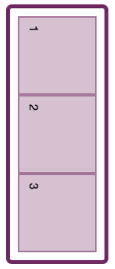
FLEXBOX CONTAINER CHEATSHEET

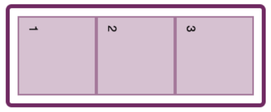
HTML	Other CSS ITEMS	Flexbox container CSS	How It Looks

<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
	* {box-sizing: border-box;}
	
	.item {
	  width: 100px;
	  height: 100px;
	  padding: 10px;
	  background-color: rgba(111,41,97,.3);
	  border: 2px solid rgba(111,41,97,.5);}
	
	.item ul {
	  margin: 0;
	  padding: 0;
	  list-style: none;}
		.container {
		  border: 5px solid rgb(111,41,97);
		  border-radius: .5em;
		  padding: 10px;
		  display: flex;
			display:flex;
			

		.container {
		  border: 5px solid rgb(111,41,97);
		  border-radius: .5em;
		  padding: 10px;
		  display: flex;
		flex-direction: row-reverse;}
			display: flex;
			flex-direction: row-reverse;
			

		.container {
		  border: 5px solid rgb(111,41,97);
		  border-radius: .5em;
		  padding: 10px;
		  display: inline-flex;}
			display: inline-flex;
			

		.container {
		  border: 5px solid rgb(111,41,97);
		  border-radius: .5em;
		  padding: 10px;
		  display: flex;
		  flex-direction: row;
		  writing-mode: vertical-lr;}
			display: flex;
			flex-direction: row;
			writing-mode: vertical-lr;
			

		.container {
		  border: 5px solid rgb(111,41,97);
		  border-radius: .5em;
		  padding: 10px;
		  display: flex;
		  flex-direction: column;
		  writing-mode: vertical-lr;}
			display: flex;
			flex-direction: column;
			writing-mode: vertical-lr;
			

Media Queries – some details

1. **@import** is used to get one style sheet to load all or part of another style sheet.

   Because it prevents parallel loading of style sheets, it slows down the page load, and **should be avoided if possible.** In other words, it is better to have (two) links to (two) stylesheets.

   @import goes right at the top of the head, just after the charset tag.

2. Firefox supports on two media types – screen and print (and FullerScreen for projection). It will work better if you use **media queries.**

   On the other hand, your aural and braille users won't be using Firefox anyway (since it doesn't support them adequately), so **@media** should be used for them.

3. Of course, your can have the media queries **inside a link to an external style sheet**.

   ```
   <link rel="stylesheet" type="text/css" media="only screen and (max-
   device-width: 780px)" href="max-device-width-780px.css" />
   ```

   or **within a style sheet**---

   ```
   <style>

   @media only screen and (max-device-width: 780px) {
         special styling goes here
         }
   general styling

   </style>
   ```

4. There is standard code for breakpoints at http://css-tricks.com/snippets/css/media-queries-for-standard-devices/

5. Next, there is excellent advice about the use (and not over-use) of breakpoints at http://bradfrost.com/blog/post/7-habits-of-highly-effective-media-queries/
   That said, you will need to use them.

6. There is a breakpoint plug-in for jQuery, with a very clear description at https://github.com/hejmartin/jquery-breakpoint (also accessible from http://plugins.jquery.com/breakpoint/)
   Basically, the plug-in monitors some condition and changes things when that condition

becomes true or false. This could be very useful if, for example, you are monitoring the orientation of a tablet or smartphone. Start at http://plugins.jquery.com/breakpoint/ and follow the links at the right side to the documentation and download.

There is also a breakpoints-js plug-in which adds classes to your elements based on breakpoints. See http://plugins.jquery.com/breakpoints-js/   and description at https://github.com/reusables/breakpoints.js

7. Many of the grid systems have code for breakpoints – e.g. responsivedesign has it at http://responsivedesign.is/develop/browser-feature-support/media-queries-for-common-device-breakpoints

8. Sass and mixins provide more structure for using breakpoints – see and
http://www.sitepoint.com/managing-responsive-breakpoints-sass/
http://responsivedesign.is/develop/getting-started-with-sass or google:
sass breakpoint tutorial
if you are interested.