**Lesson 5 - CSS: Height, Width, Box-Sizing, Positioning**

**Lecture Notes**

**Activity - All about size**

1. Google Chrome Extension - Window Resizer
2. How big is a certain device? <http://mydevice.io/devices/>
3. <http://whatismyviewport.com/> (useful for checking what your viewport is on any device)
4. Google chrome developer tools (mobile button) changes device mode

**Lecture**

1. **CSS Height and Width Dimensions**
   1. Suggested Reading

<http://www.w3schools.com/css/css_dimension.asp>

* 1. Setting height and Width
     1. Height and Width are set to “auto” by default
     2. Auto means browsers calculate the height and width for you
     3. You can set to pixels (i.e. 20px), or in percentages (i.e. 100%)
  2. Height and width are the last piece of the box model.
     1. Margin -> Border -> Padding -> Height/Width
  3. Max-width
     1. Overrides the property “width” (more powerful)
     2. When the browser gets smaller than the max width, the width will scale down

|  |  |
| --- | --- |
| **Property** | **Description** |
| [height](http://www.w3schools.com/cssref/pr_dim_height.asp) | Sets the height of an element |
| [max-height](http://www.w3schools.com/cssref/pr_dim_max-height.asp) | Sets the maximum height of an element |
| [max-width](http://www.w3schools.com/cssref/pr_dim_max-width.asp) | Sets the maximum width of an element |
| [min-height](http://www.w3schools.com/cssref/pr_dim_min-height.asp) | Sets the minimum height of an element |
| [min-width](http://www.w3schools.com/cssref/pr_dim_min-width.asp) | Sets the minimum width of an element |
| [width](http://www.w3schools.com/cssref/pr_dim_width.asp) | Sets the width of an element |

1. **Box Sizing**
   1. **Suggested Reading**

<http://learnlayout.com/box-sizing.html>

<http://www.paulirish.com/2012/box-sizing-border-box-ftw/>

<http://www.w3schools.com/cssref/css3_pr_box-sizing.asp>.

<http://blog.teamtreehouse.com/box-sizing-secret-simple-css-layouts>

* 1. **Content-Box**
     1. Default. The height and width properties include ONLY the content. Border, Padding, Margin are not included
     2. Can be confusing/annoying when you set width and then the your element is too wide because of padding/border
     3. Makes code less readable
  2. **Border-box**
     1. We are going to be using border-box for everything.
     2. Any padding (or border) you set will not be added to the rendered width.
     3. Width controls width.
     4. Easier to read code

1. **Position**
   1. In order to make more complex layouts, we need to discuss the position property. It has a bunch of possible values, and their names make no sense and are impossible to remember. Let's go through them one by one, but maybe you should bookmark this page too <http://learnlayout.com/position.html>
   2. **Static**
      1. The default value
      2. Everything has static positioning.
      3. static position = not positioned
   3. **Relative**
      1. Behaves just like static
      2. You can “nudge” it using top/bottom/left/right
      3. It will appear where it normally will, but adjusted by just a few pixels.
      4. Very useful for manually getting stuff to line up when nothing else works.
      5. Padding, margin, text-align, etc.. not working? Use relative position.
   4. **Fixed**
      1. Affixed to the page. Will not move even if you scroll.
      2. Someone took a hammer and nail and attached it to the browser window
   5. **Absolute**
      1. absolute is the trickiest position value. absolute behaves like fixed except **relative to the nearest positioned ancestor** instead of relative to the viewport. If an absolutely-positioned element has no positioned ancestors, it uses the document body, and still moves along with page scrolling. Remember, a "positioned" element is one whose position is anything except static.
      2. <http://learnlayout.com/position.html>
2. **Float**
   1. For now, just assume that it aligns divs either left or right. We’ll do another lesson on float, clear, clearfix, etc…