PART ONE

=================

There are a few key points from Chapter 13 - Boxes. Here are some of them:

Width: use px or %

Height: use px or %

example:

div {  
height: 300px;  
width: 300px;  
background-color: #bbbbaa;

}

When using percentages (%) vs. px, your boxes will be more responsive, as they change with the re-sizing of your window when it;s enlarged or shrunken down. The issue with this, is that it's a bit difficult to predict what boxes will look like with resizing. For resizing your boxes to be "predictable"  and following good responsive practices, use "**@media**" and **px** and rewrite the size of your boxes according to your "**conditions**" (i.e **@media (min-width: 481px) and (max-width: 768px)** )

**Overflow** is great for boxes that have a lot of text but you don't want the text to take up your entire page:

p {  
overflow: scroll;

}

Border, Margin, and Padding -- The two that you will mainly work with are Margin and Padding:

(see page 308 for a good example)

**Margin**: Pushing boxes around "externally" on your main page -- essentially it allows you to move your box around the entire web page

**Padding**: Moving content INSIDE your boxes (examples is are they flush again the inner wall of the box -- or do they have a little bit of space between the inner edge of the box and themselves?)

**Border**: We use this at the moment to see how big our boxes are -- use this to so that you can "see" where you're boxes are being placed. When you change the size of your border, be sure to use the "box-sizing" trick at the top of your CSS file which is:

**\* {**  
**box-sizing: border-box;**  
**}**

(do remember that the top code DOES NOT cover psuedo elements as seen on page 289 - which can be very useful! If you want to make sure they are protected from weird sizing issues, use this syntax below instead:

**\*, ::before, ::after {**  
**box-sizing: border-box;**  
**}**

(Single colon ":" is more compatible with older browsers, and what you see above is actually a double colon "::"  then the word -- use commas between each element to make sure they are separated)

*Here is some more info on the box-sizing trick:*

<https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>

[Links to an external site.](https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing)

For Border Width, Style, and Color -- we use the shorthand way of writing it:

**border: width style color;**

which looks like this below in context:

**border: 2px solid black;**

If you want to specify a specific side of the border, use the shorthand like this (for example, top and bottom of the border):

**border-top:** **2px solid black;**

**border-bottpm: 2px solid black;**

With the code above, we will not have sides on our on border, only the top and bottom of the border will appear.

**display: inline;** This is will turn block-elements into inline-elements

**display: block;** This will turn inline-elements into block-elements

**display: inline-block;** Turns elements into inline-elements AND allows the use of block-elements properties+values.

For getting a clearer picture of *display: inline-block;* please visit this page:

<https://www.w3schools.com/Css/tryit.asp?filename=trycss_inline-block_span1>

[Links to an external site.](https://www.w3schools.com/Css/tryit.asp?filename=trycss_inline-block_span1)

**display: none;** The element that would be there is hidden AND it's placement is no longer recognized as seen on page 317

**visibility: hidden;**The element will disappear, but its placement on your page will remain (there will be an empty space where the element should be).

**Box-shadows** are great and these two are great examples of its use:

box-shadow: 5px 5px 5px 5px #777777;

box-shadow: 0 0 10px #777777;

Border-radius will be a big one to use for styling purposes. I advise getting very familiar with this syntax. Here is a good example of code:

border-radius: 10px; (all around -- which starts and goes from the top, right, bottom, left -- make note that it moves "clockwise")

border-radius: 10px 60px;

(top left, bottom right will be 10px)

(top right and bottom left will be 60px)

PART TWO

=================

**Lecture for Chapter 13 (section 2) - Media Querys, Repsonsiveness, and Box-Sizing**

This information is only available via this lecture (not found in book!):

<https://youtu.be/ZKlah9sOsyU>

[Links to an external site.](https://youtu.be/ZKlah9sOsyU)

**List of sizing for "responsiveness" or "responsive web sites"**

320px — 480px: Mobile devices  
481px — 768px: iPads, Tablets  
769px — 1024px: Small screens, laptops  
1025px — 1200px: Desktops, large screens  
1201px and more —  Extra large screens, TV

Keep in mind that we will only use Mobile and iPad sizing (watch video for more detail) or use numbers below:

REFERENCE for @media query sizing:  
480px: Mobile devices  
481px — 768px: iPads, Tablets  
(Regular sizing doesn't need a @media query)

To read more about @media queries:

<https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/#:~:text=Normally%2C%20the%20text%20size%20will%20be%2014px.%20However,queries%20at%20the%20end%20of%20your%20CSS%20file>

[Links to an external site.](https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/#:~:text=Normally%2C%20the%20text%20size%20will%20be%2014px.%20However,queries%20at%20the%20end%20of%20your%20CSS%20file)

Read more on box-sizing here:

<https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>

[Links to an external site.](https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing)

Syntax to memorize below:

\*, ::before, ::after {  
    box-sizing: border-box;  
}

More info on

display: inline;

display: block;

display: inline-block;

<https://www.w3schools.com/Css/tryit.asp?filename=trycss_inline-block_span1>