HTML Notes in General (1st Lesson)

HTML is made up of references to other documetns

Links to another pages

Markups are made of tags

Tag-content-tag = Element

em = emphasis -> italic

<span> is inline and div is block

The <a> stands for Anchor Tag

The BR its tricky, it seems like a block tag, but doesnt make a box on the screen

What was the "strong" tag? (I'm getting back here). Strong makes bold the whole sentence, while b tag is used to bold a part of the sentence.

And what is "Form" tag? (I searched this in w3 Schools) The <form> tag is used to create an HTML form for user input.

CREATING A STRUCTURED DOCUMENT WITH HTML (2D LESSON)

(Copied from Instructor Notes)

Box Sizing

There are four main points that Jessica addressed about box sizing.

HTML elements are boxes and each box has 4 components.

Because there are so many components to each box, it can often be hard to get the size of a box just right.

There are two techniques you can use to help deal with sizing issues.

Set sizes in terms of percentages rather than pixels.

Set the box-sizing attribute to border-box for every element (I need to try more examples of this)

Different browsers work slightly differently. Sometimes this causes different browsers to display the same code differently.

Box Positioning

Divs are block elements (as opposed to inline), so by default they take up the entire width of a page.

Adding the rule display: flex; to the appropriate CSS will override this behavior and let divs appear next to each other.

(Took from w3 Schools)

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to add a border around elements, and to define space between elements.

ADDING CSS STYLE TO HTML STRUCTURE (3RD LESSON)

Always on the Head tag should we use the CSS tags or styling? I'm getting back here... Yes, though I tested using it outside or even outside <html> it works, but its incorrect since it does not comply with the HTML structure.

Ok, you havve 2 ways of working or inserting style to your HTML document.-> YOu can just add <style> tag or add the <link> tag.

In the first one, you set all the attributes you need, then you start with the <head> tag, so every code you write after that, becomes stylished. (I'm getting back here. I was stating this after several days of not touching my project, and this is what I remembered. Now I see its wrong, well partially wrong).

The second one is to relate your HTML to another document, which in this case will be a css doc, where you will set all the attributes you need and then just make reference to it, stating that same document as a whole and the using the parts you need with the class=xxx tag.

Three Ways to Insert CSS (I researched this on w3 school)

There are three ways of inserting a style sheet:

Inside HEAD

External style sheet-> &lt link rel="stylesheet" type="text/css" href="mystyle.css" &gt

Internal style sheet-> Internal styles are defined within the style element, inside the head section of an HTML page:

Inline style

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

(I tested later that it doesnt matter where you put the <styles> or the links, it will work, though its totally incorrect)

HTML tag -> Use an HTMl tag in css to intruct the program to use that style on all those tags....OR

Use the the CLASS tag in HTML and then describe it in CSS..eg : .description --- .content

(Instructor Notes)

Method 3: Write your style inline with your HTML

Example

<body>

<div style="background-color: red; color: white">

This div will have a red background and white text.

</div>

<div style="background-color: red; color: white">

So will this one.

</div>

<div style="background-color: red; color: white">

Now, what if I change my mind?

</div>

<div style="background-color: red; color: white">

I'd rather have a black background...

</div>

<div style="background-color: red; color: white">

Never do this!

</div>

</body>

RANDOM NOTES:

Where to write code for practice:

codepen.io

scratchpad.io

Text Editors:

Sublimetext2

Notepad

Notepad++

(I copied this from instructor notes. This explains an important part of my interest in learning programming)

Thinking like a Programmer

A large goal of this Nanodegree is to give you practice thinking like a computer programmer. Longtime programmers will often talk about how the programming thought process has helped them in other areas of their life, but what does it mean to "think like a programmer?" In this Nanodegree we will focus on five specific ways of thinking that all programmers have to use all the time. We will come back to these ideas repeatedly throughout the Nanodegree.

1. Procedural Thinking

2. Abstract Thinking

3. Systems Thinking

Systems thinking happens when you break a big problem down into smaller pieces. Programmers do this when they create a plan (often on paper) for how a program will work. It involves big-picture thinking and decision-making about a problem and how different pieces of a program can work together to solve it. If this is unclear now, don't worry! You'll understand what this means much more clearly by the end of the Nanodegree.

4. Technological Empathy

5. Debugging

Collect evidence (what makes this program fail?)

Generate theories (what may have caused this problem?)

Test those theories (if my theory is correct, how could I find out?)

Fix the problem

WHERE TO VERIFY YOUR HTML AND CODES

To verify HTML: <http://validator.w3.org/#validate_by_input>

To verify CSS: <http://jigsaw.w3.org/css-validator/#validate_by_input>