CINS 110 Notes

Blocks:

I think that is the easiest to begin thinking about blocks as actual objects on a webpage. Block elements start a new line and take the space of the entire page from the far left to the far right. The block-level elements include ** <h1> - <h6> <form>.** So if you were to introduce one of these on the document it would look like this.

<h1 ALIGN="center"> This is Assignment: 04x </h1>

This is Assignment: 04x

↑ So you can notice here that it takes up all this space from both sides of the page ↑

Inline:

Inline elements are style elements that only take up as much width as necessary. They don't take the whole page unless you want it to. Inline elements include ** <a> <img>.** Inline would look something like this.

 This is Assignment: 04x

This is Assignment: 04x

Entities:

From what I understand entities are used to if you want to display the reserved characters. In HTML we have to use <> to denote tags and other characters like # and & for special characters so if you want to actually use a "&" on your page you have to use an entity to display the character. So for example & is the ampersand would print "& is the ampersand". Or if you wanted you could use & instead. Either way entities can be used by their name preceded by the & or by their number preceded by the &.

UTF-8:

Similarly to Entities, UTF-8 characters are displayed using a & like an entity. UTF-8 symbols are pretty cool and can summon sweet special characters depending what browser you are using. According to w3 schools IE and firefox have the best support. Although there are not many icons supported in HTML if you want to include cool icons on your page you can use CSS.

After about 30 minutes of google help you can find many cool sites that offer kits and packs of special characters that are free to use in CSS code, even commercially! They require you to include the path to the .min.css stylesheet that includes these icons and logos. An example at the top of your HTML document would look something like this.

```
<link rel="stylesheet" href="path/to/*.min.css">
```

Some Icons I will include below that come from this special kit.

This can be used for the "settings" page of your site.

This is a cool little printer logo

This is a Rebel Alliance logo from Star Wars to spice up your site!

CSS Selectors:

A CSS selector allows you to name many elements when you are trying to style them. For example on the style sheet you could say

div, p, h1 { color: red }

This would change the color for all of these elements.

CSS Units:

A CSS unit is essentially a measure of the size of an element. The first diagram is an example of relative unit sizes and the bottom are the absolute unit lengths.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

CSS Colors:

Colors come in 3 flavors. Named, rgb value, and hex. An Example of red in all 3 would be

-red

-rgb(255,0,0)

-#ff000

-FF000

CSS Syntax:

The selector points to the HTML element you want to style.
The declaration block contains one or more declarations separated by semicolons.
Each declaration includes a CSS property name and a value, separated by a colon.
A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
The Syntax should look something like this
p {background-color: red; color: black;}

```
div > p {
    background-color: yellow;
}
```

CSS Combinators:

A CSS selector can contain more than one simple selector.
Between the simple selectors, we can include a combinator.
There are four different combinators in CSS3:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

```
div + p, div > p {
    background-color: yellow;
}
```

The descendant selector matches all elements that are descendants of a specified element.
The child selector selects all elements that are the immediate children of a specified element.
The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.
Sibling elements must have the same parent element, and "adjacent" means "immediately following".
The general sibling selector selects all elements that are siblings of a specified element.

CSS Psuedo-Classes:

A pseudo-class is used to define a special state of an element.
For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

```
selector:pseudo-class {
    property:value;
}
```

```
/* mouse over link */
a:hover {
    color: #FF00FF;
}
```

CSS Psuedo-Elements:

A CSS pseudo-element is used to style specified parts of an element.
For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

```
p::first-letter {
    color: #ff0000;
    font-size: xx-large;
}
```

You can use the :

```
::-moz-selection { /* Code for Firefox */
    color: red;
    background: yellow;
}

::selection {
    color: red;
    background: yellow;
}
```

This is some text in a div element.