CSS stuff-1
```css
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: pink;
}

h1, p {
    color: white;
}

p {
    font-family: verdana;
    font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: yellow;
}

h1, p {
  color: white;
}

p {
  font-family: verdana;
  font-size: 20px;
  color: green;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<style>
.intro {
  background-color: yellow;
}
</style>
</head>
<body>
<h1>Welcome to My Homepage</h1>
<div class="intro">
  <p>My name is Donald.</p>
  <p>I live in Duckburg.</p>
</div>
<p>My best friend is Mickey.</p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
    <style>
        .intro {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <h1>Welcome to My Homepage</h1>

    <div class="intro">
        <p>My name is Donald.</p>
        <p>I live in Duckburg.</p>
    </div>

    <p>My best friend is Mickey.</p>
    <p class="intro">My best friend is Mouse.</p>
</body>
</html>
Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The `<div>` element is a block-level element.

Example

```html
<div>Hello</div>
<div>world</div>
```

```html
<!DOCTYPE html>
<html>
<body>
  <div>Hello</div>
  <div>world</div>

  <p>The DIV element is a block element, and will start on a new line.</p>
</body>
</html>
```
<!DOCTYPE html>
<html>
<body>
<div>Hello</div>
<div>World</div>
<p>The DIV element is a block element, and will start on a new line.</p>
<p>Hello</p>
<p>Class</p>
</body>
</html>
Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all `<p>` elements inside `<div>` elements:

Example

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

Try it Yourself »
<!DOCTYPE html>
<h1> p m </h1>
<!DOCTYPE html>
<html>
<head>
<style>
  div p {
    background-color: yellow;
  }
</style>
</head>
<body>
<h1>blah blah blah</h1>
<p>di di di</p>
<h1>hi helen!</h1>
<p>da da da</p>
<h1>HELEN</h1>
<p>Paragraph 1 in the div.</p>
<p>Paragraph 2 in the div.</p>
<p>Paragraph 3 in the div.</p>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
</body>
</html>
CSS SYNTAX

selector {
  property1: value1;
  property2: value2;
  
  
}

these are associated with HTML tags

THE RULES ASSOCIATED WITH THE SELECTOR WILL BE APPLIED TO ALL INSTANCES OF A TAG ON THAT PAGE.

EXAMPLE:

p { color: blue; }
li { color: red; }

FILE STRUCTURE:

cogsDemo
  my_css.css
  my-page.html

<link rel="stylesheet" href="my_css.css" type="text/css" />

<html>
  <head>
    <link href="my_css.css" />
  </head>
  <body>
    <p> this is a cool demo in COGS3</p>
  </body>
</html>
more than one selector at a time

eg

```css
p, ol, ul { color: blue; }
ol { color: blue; }
ul { color: blue; }
```

comma separated list

p, ol, ul { color: blue; }

contextual selectors

*These are used to apply styles to elements only when they are nested within other specified elements.*

```css
p ol li { color: pink; }
```

notice, no comma!

this means that the rule only applies to li element nested in ordered lists.
example: CSS specificity

```
cite { color: green; }
p cite { font-style: italic; font-weight: normal; color: red; }
li cite { font-style: normal; font-weight: bold; color: blue; }
```

Voila! you can override property settings with more specific selectors.

This rule applies to all cite tags.

Nested styles will override the default cite tag because they are a more specific style definition.

CSS CLASSES

Creating your own classifications

Any element with this class will follow this rule.

**HOW TO CREATE A CLASS**

Notice the leading " . "

CSS IDs

Used to assign identifiers to unique elements

Style.css

```
#cool {
  color: pink;
  font-weight: bold;
}
```

Cool.html

```
<div id="cool">some text</div>
<h1 class="cool">un <h1>
```
If you want to have this class only have a class of 4 x's apply to 4 tags.
Creating Tables

- All components of a table are placed in here.

```
<table>
  <tr>
    <td>...<td>
    <td>...<td>
    <td>...<td>
  </tr>
</table>
```

- This is a table cell.

- The table structure is defined by rows.
- The caption defines the name of a table.

```
<table>
  <caption>MY TABLE</caption>
  <tr>
    <th>heading1</th>
    <th>heading2</th>
    <th>heading3</th>
  </tr>
  <tr>
    <td data1</td>
    <td data2</td>
    <td data3</td>
  </tr>
</table>
```

- Usually the first row of the table.
- Normally in cell.
- Left justified.

- The caption is centered.
- To get a border for your table:

```
<table border="1"/>
```
**USING STYLES WITH TABLE**

```css
/*table*/
    border: 1px solid black;
    border-collapse: collapse;
*/

```