

HTML & CSS

comment in HTML:

```
<!--  
inside the comment  
-->
```

Change color of text: HTML

```
<h2>CatPhotoApp</h2>
```

to:

```
<h2 style="color:blue">CatPhotoApp</h2>
```

you can also use hex or html or rgb

ie) rgb(255, 255, 255)

abbreviated hex:

Color	Short Hex Code
-------	----------------

Cyan	#0FF
------	------

Green	#0F0
-------	------

Red	#F00
-----	------

Fuchsia	#F0F
---------	------

Change color of text: CSS

```
<style>
```

```
h2 {
```

```
  color:blue;
```

```
}
```

```
</style>
```

```
<h2>CatPhotoApp</h2>
```

you can also use hex or html

Change color using classes: CSS

```
<style>
```

```
  .red-text { // declare class red-text
```

```
    color: red;
```

```
  }
```

```
<\style>
```

```
<h2 class="red-text">Test</h2> // now the h2 has a class of "red-text"
```

```
<p class="red-text">Kitty ipsum dolor sit amet, shed everywhere shed everywhere
```

```
stretching attack your ankles chase the red dot, hairball run catnip eat the grass sniff.</
```

```
p> //p also has class of red-text and is now red
```

you can also use hex or html

Change font / font-family / anything: CSS

```
<style>
```

```
p {
  font-family: Monospace;
  font-size: 16px;
}
</style>
<h2 class="red-text">CatPhotoApp</h2>
```

```
<p class="red-text">Kitty ipsum dolor</p> // size 16px
<p>Purr jump the sink fluffy fur catnip scratched.</p> // size 16px
```

Import Google Font:

```
<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">
<style>
  h2 {
    font-family: Lobster, Monospace; // if the first font isn't working for some
                                     reason it will degrade to the next one
  }
</style>
<h2 class="red-text">CatPhotoApp</h2>
```

```
<p class="red-text">Kitty ipsum dolor</p> // size 16px
<p>Purr jump the sink fluffy fur catnip scratched.</p> // size 16px
```

Images in HTML:

img src gives the image source
alt gives an alternate if the image can't appear for any reason
``

Change image size: refer to class, basically same thing i.e.

within style:

```
.smaller-image {
  width: 100px;
}
```

and then apply to the image before the src

Add border to anything: CSS

In `<style>`:

```
.thick-green-border {
  border-color: green;
  border-width: 10px;
  border-style: solid;
}
```

and then call it as a class to anything you want to make into a border

NOTE: you can also add multiple classes if you want
ie:

```

```

Round out borders: CSS

```
border-radius: 10px;  
//use this in any class declaration  
border-radius: 50%  
// rounds out image and only shows 50% from center
```

Embedded Link: HTML

```
<p>Click here for <a href="http://www.freecatphotoapp.com">cat photos</a></p>
```

<a> anchor tags and anchor text surrounded by a p tag and text

Creates: Here's a [link to Free Code Camp](#) for you to follow.

if you replace the href link with a # it will make it a dead link

*NB you can also nest an image in the text part of your anchor tag for a clickable image, if necessary

Unordered List: HTML

```
<ul>  
  <li>a</li>  
  <li>b</li>  
  <li>c</li>
```

```
</ul>
```

ul= unordered list

li= list item

Ordered List: HTML

```
<ol>  
  <li>dogs</li>  
  <li>other cats</li>  
  <li>water</li>
```

```
</ol>
```

ol= ordered list

li= list item

Accept input from textbox:

```
<input type="text" placeholder="cat photo URL.">
```

placeholder is a placeholder before text is entered

Submit information to a server through text fields: HTML

```
<form action="/url-where-you-want-to-submit-form-data">
```

```
  <input type="text" placeholder="cat photo URL.">
```

```
</form>
```

this will send what is in-between the 2 form tags to the server. If you place a input tag between, the inputted text will be sent to the server

***NB: Button in form action**

If you insert a button type into your form action, then you can also have a button
Final result should look like this:

```
<form action="/submit-cat-photo">
  <input type="text" placeholder="cat photo URL">
  <button type="submit">Submit</button>
</form>
```

***NB: Requiring input**

If you add a required key word to the end of your input tag, it will make sure you have to submit, but this doesn't work with safari:

```
<input type="text" placeholder="cat photo URL" required >
```

Submit info through radio buttons: HTML

```
<form action="/submit-cat-photo">
  <label><input type="radio" name="indoor-outdoor">Indoor</label>
  <label><input type="radio" name="indoor-outdoor">Outdoor</label>
  <button type="submit">Submit</button>
</form>
```

*all forms should have button at the end
make sure to add this to your form tag with all your other stuff
inside label tag
make an input tag with a type="radio"
name it the options
and the labels

(adding checked at the end of any tag both radio and checkbox marks it default as checked)

Submit info through checkbox buttons: HTML

Radio buttons can't be deselected if selected once, checkbox types can

```
<form action="/submit-cat-photo">
  <label><input type="checkbox" name="personality" checked> Loving</label>
  <label><input type="checkbox" name="personality"> caring</label>
  <label><input type="checkbox" name="personality"> heartfelt</label>
  <button type="submit">Submit</button>
</form>
```

make sure to keep the name the same for related attributes

(tags that don't need a (/) closing tag are called self closing)

div element: HTML

div tags are tags that are used to either group or divide groups of text in HTML. It is the highest cardinality (compiled last) and has the most weight when making style changes:

```
<div style="color:#0000FF">
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
```

***NB: adding classes to div element**

```
<div class="silver-background">
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
```

id attribute: HTML

add to the end of any tag so that it will be easier in jQuery to find it

id attribute: CSS

```
#cat-photo-form {
  background-color: green;
}
```

it makes all id's have same style attributes

Change Box size of attribute: CSS

```
.green-box {
  background-color: green;
  padding: 20px;
  margin: 20px;
}
```

padding is the vertical height from last element

margin is width of element not related to any internal tag length

***NB: negative margin**

increases the size of element

***NB: internal element overflow**

if an internal element dimensions are larger then its wrapped element, the wrapped element will go away

***NB: Individual side PADDING: CSS**

padding-top, padding-right, padding-bottom, and padding-left (internal size)

***NB: Individual side MARGIN: CSS**

margin-top, margin-right, margin-bottom, and margin-left (external size)

***NB: Individual side (single line) Margin & padding: CSS**

```
padding: 10px 20px 10px 20px;
      (top) (right) (bottom) (left)
```

Body element:(entire screen)

classes applied to element in HTML reference are compiled last and have highest cardinality

ie:

```
<style>
.pink-text {
  color: pink;
}
.blue-text {
```

```
    color: blue;
}
#orange-text {
    color: orange;
}
</style>
<h1 id="orange-text" class="pink-text blue-text">Hello World!</h1>
/////HTML always goes with whats last, the text will be orange because id tags are
compiled later
if it were:
<h1 id="orange-text" class="pink-text blue-text" style="color:green">Hello World!</h1>
/////then it would be green because inline style attributes are compiled later

*NB: important keyword:CSS
if you have any attribute with an important tag like this:
.pink-text {
    color: pink !important;
}
```

it will always go to that first regardless of cardinality