

Tuesday



Objective 2: give it style with CSS

- CSS: Cascade StyleSheet → the last styling option is the one kept
- → if you put `p {color:red;}` then `body {color:blue; font-style:italic;}`

And you have:

```
<body>
```

```
<p> Paragraph </p>
```

```
</body>
```

In your html: what will you get?

Paragraph or *Paragraph*?

Objective 2: give it style with CSS

- 3 ways to include style:
 - Either between style tags → `<style></style>`
 - In an external file (my preferred method):
 - `<link rel="stylesheet" href="style.css">`
 - This goes in the head
 - Style each item individually:
 - `<p style="color:red; font-style:bold;"> Paragraph </p>`

Objective 2: give it style with CSS

- Generally:

```
type {  
  attribute-to-change: value;  
}
```

Ex:

```
h1 {
```

→ Changes EVERY h1

```
  text-align:center;  
  of division it is in  
}
```

→ Sets the alignment to center (like <center> tags)

Objective 2: give it style with CSS

- This poses a problem: what if I wanted to change 1 h1 but not another? (or a p, div, img etc)
- Solution: classes and ids
- Difference: class should be used for several objects (see it as a category) and id should be used for a single object (will be seen again when doing javascript)

Objective 2: give it style with CSS

- When styling IDs:

`#id-name {attributes...}`

- When styling classes:

`.class-name {attributes...}`

Ex: see example page

Objective 2: give it style with CSS

- Selectors:

tag.class → selects all tags that have that specific class

tag1, tag2, tag3... → apply this to all of tag1 and tag2 and tag3

Objective 2:

- A few widely used attributes:

`border: [size] [color] [style]`

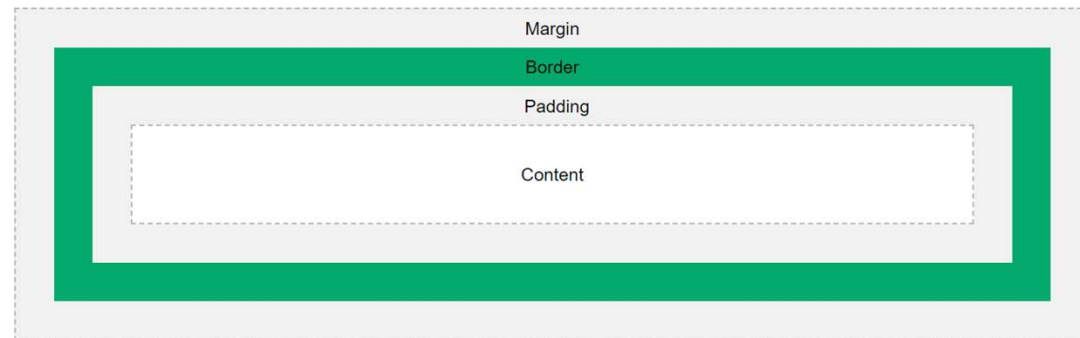
Ex: `border: 2px outset lightcoral` (see example page h1)

`padding, margin: [size]`

padding is space between content and border, margin is space between border and other item

`background-color: [color]`

`background-image: [see example page]`



Objective 2:

- Color: [font-color]
- Opacity: [opacity percentage, 100% is all normal, 0% is transparent]
- Font-style: [italic, oblique, normal, inherit]
- Text-decoration: [dashed, dotted, double, line-through, none, overline]
- ::selection {attributes} → apply style to text when it's selected
- Item:hover {attributes} → apply style to item when the mouse is over it (similar types of selectors include ::first-letter, :link, :active, :visited, :first-line etc)
- Display: {attributes} → display an inline as a block or vice versa, you can use none to not show an object (make it disappear)

Objective 2:

- Setting 2 (or more) items next to each other (the pikachu images)
- Best solution:
 - Create a div to store everything (don't need to give it a border or background color)
 - Set both items with a combined width + margin + padding + borders < 100%
 - Give them each a class
 - In CSS (tags or external stylesheet): set float to either right or left
 - Give the div an overflow: auto or hidden

Float: left [or right] → send the next item to the left [or right]

Objective2:

Reference:

<https://www.w3schools.com/cssref/default.asp>