github.com/devDogra

Go to the ‘films’ repository and copy index.html

3 ways of adding CSS

1) **Inline:** using style attribute

   <p style=”color: blue” > This is a para **</p>**

2) **Internal:** inside <style> tag
3) **External:** inside a new ***.css*** file

   Add <link href=”styles.css” rel=”stylesheet”> to the HTML <head> tag

   To link the .css file to our HTML file

CSS Selectors

- We select elements using selectors, and then apply styling to them. ***ALL = CSS rule***

If 2 rules with same selectors, the last one is used

Some properties (use MDN for reference)

- color
- background-color
- font-size
- font-weight
- border

Multiple selectors

```
h1, h2, h3 {
    text-transform: uppercase;
}
```

Devtools and CSS

How to select specific elements?

- ID selectors

  HTML: Hi this is a para

  CSS:

  #verification { .... }

How to select a group of specific elements?

- Class selectors

  HTML: My para

  CSS:

  .my-para { ... }

  (notice the dot)

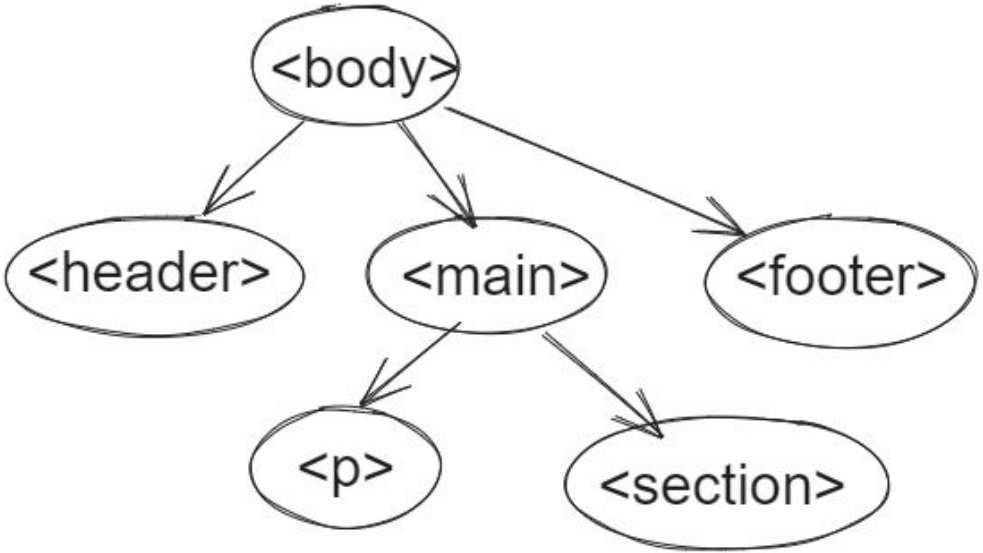
Select elements with multiple classes

```
.class1.class2 {
    .......
}
```

(no space between .class1 and .class2)

Elements are structured like a tree

All text properties are inherited



. . . . . and so on

Chaining selectors

Descendent selector: div p { … }

Next adjacent sibling: li + li { … }

General sibling: li ~ li { … }

Specificity

```
<div id="parent-id" class="parent-class">
    <div id="child-class">
        <p> Hello Hi</p>
    </div>
</div>
```

Which rule will be applied?

#parent-id p { … }
vs
.parent-class p { … }
vs
.parent-class > .class-child > p { … }

Specificity of a selector

Id in selector = 100 points

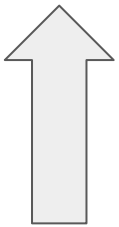
Class, attribute = 10 points

Element = 1 point

.parent-class > .class-child > p
Has 2 class selectors, and 1 element selector, so

2*10 + 1 = 21 points

#parent-id p
Has 1 id selector, and 1 element selector
So
100 + 1 = 101 points



MORE SPECIFIC, so will be applied

Inline styles override all other styling

!important has the highest specificity, even more than inline

pseudo-classes

:hover, :visited etc

li:nth-child(2n + 3)

=> Selects all s that are a 5th (n=1) child, 7th (n=2) child, 9th (n=3) child and so on…