Module 3-2

CSS Selectors

HTML Elements: The Box Model

- HTML content that have been annotated with tags **are known as HTML elements**.
- All elements come with a margin, border, and padding, this is referred to as the box model.

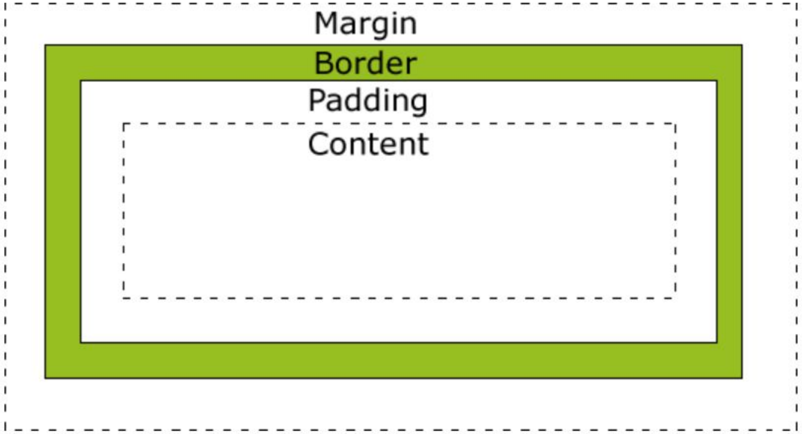


Image - Image of Box Model

HTML Elements: Inline vs Block

- HTML elements are also classified as being inline vs block.
    - **Inline**: Does not start on a new line
    - **Block**: Starts on a new line

- Common inline elements: a, img, span
- Common block elements: p, div, table

HTML Elements: inline-block

There is a third type of display known as an inline-block to which elements can be set to, which has characteristics of both inline and block elements:

- Unlike block elements, they don’t create line breaks.
- Unlike inline elements, you can adjust its width and height.

Let’s Test This

HTML Elements: Positioning

- All elements have a default flow, a position they will fall into in the absence of additional instructions. This is known as “static” flow. There are additional defined positions:
    - **relative**: “relative” to what it would be positioned per the normal flow. (Hard to explain, we’ll do an example)
    - **absolute**: positioned relative to its ancestor.
    - **fixed**: positioned relative to your screen, will follow you as you scroll.

Let’s Test This

CSS Selectors

- CSS uses selector to determine which HTML elements will be “targeted” or selected to have a specific format.
- Generally speaking, there will be a CSS block that looks like so:

```
SELECTOR {

        attribute : attribute value
}
```

- We will start discussion three most important types of selectors in the next section.

CSS Selectors: By Element

Example

```
div {

        color : red;
}
```

A valid HTML element type is used, in the example to the left, the DIV type.

<u>What this code does</u>: Finds all HTML elements that are <div>'s and applies the formatting, which is to make all the enclosed text red.

CSS Selectors: By Class

Example

```
.warning {

        color : red;
}
```

The dot is required for selection by class.

Anytime you see a dot, it is selection by class!

A valid class name is used, in the example to the left, any element of class “warning.”

<u>What this code does</u>: Finds all HTML elements that have the class specified. In other words find all HTML elements that are enclosed by any tag with an attribute class=”warning”.

CSS Selectors: By ID

Example

```
#demo {

        color : red;
}
```

The # sign is required anytime you do selection by ID.

Anytime you see a #, it is selection by id!

A valid HTML element type is used, in the example to the left, the DIV type.

What this code does: Finds all HTML elements that have an attribute id=”demo” and apply the format.

CSS Selectors: ID takes precedence over Class

Example

```
#demo {
        color : red;
}

.someId {
        color: blue;
}
```

If there is a HTML element has both an id of someId and a class of demo it will appear red.

<p id= “demo” class= “someId”>...</p>

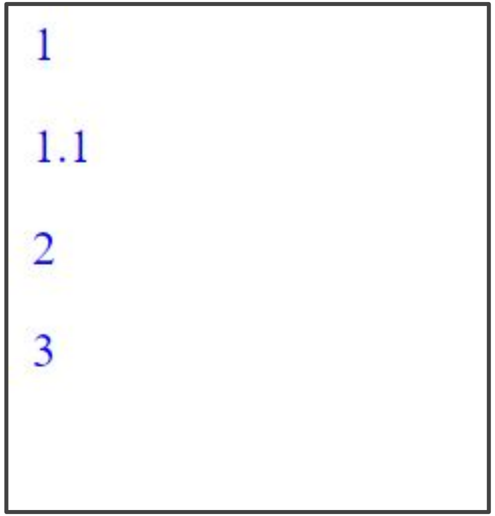
The Id attribute takes precedence over the class attribute.

CSS Descendant Selector

Consider the following CSS & HTML code:

```
div p {
        color: blue;
}
```

```
<div id="container">
   <p>1</p>
    <span>
        <p>1.1</p>
    </span>
   <p>2</p>
   <p>3</p>
</div>
```



1

1.1

2

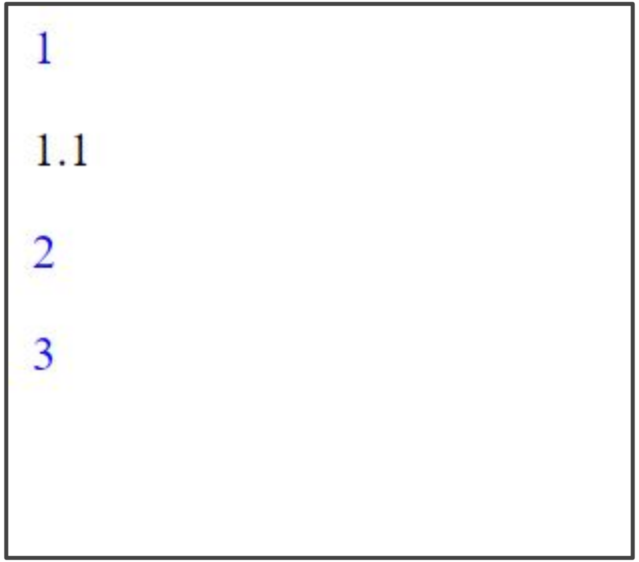
3

CSS Child Selector

Consider the following CSS & HTML code:

```
div > p {
        color: blue;
}
```

```
<div id="container">
   <p>1</p>
    <span>
        <p>1.1</p>
    </span>
   <p>2</p>
   <p>3</p>
</div>
```

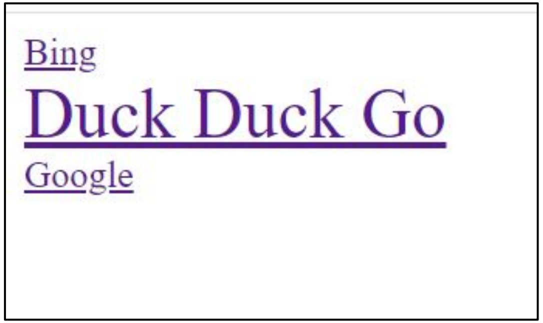


```
1

1.1

2

3
```

CSS Selector by attribute

Consider the following CSS & HTML code:

```
a[target] {
    font-size: 30px;
}
```



Bing
Duck Duck Go
Google

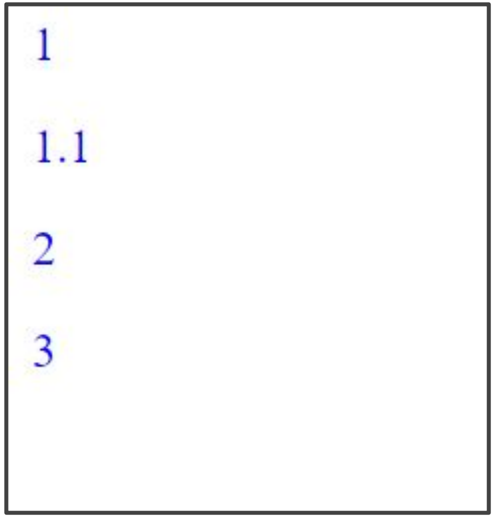
```
<html>
    <head>
        <title>A Simple Page</title>
        <link rel="stylesheet" href="style.css">
    </head>

    <body>
        <a href='https://www.bing.com' >Bing</a><br>
        <a href='https://duckduckgo.com/' target = '_blank'>Duck Duck Go</a><br>
        <a href='https://www.google.com' >Google</a>
    </body>

</html>
```

Note that there are many anchor tags, but only the one with a target attributed was affected.

Let’s write some code!

CSS Variables

```
:root {
   --main-bg-color: blue;
 }

div  p {
   color: var(--main-bg-color);
}
```



1

1.1

2

3