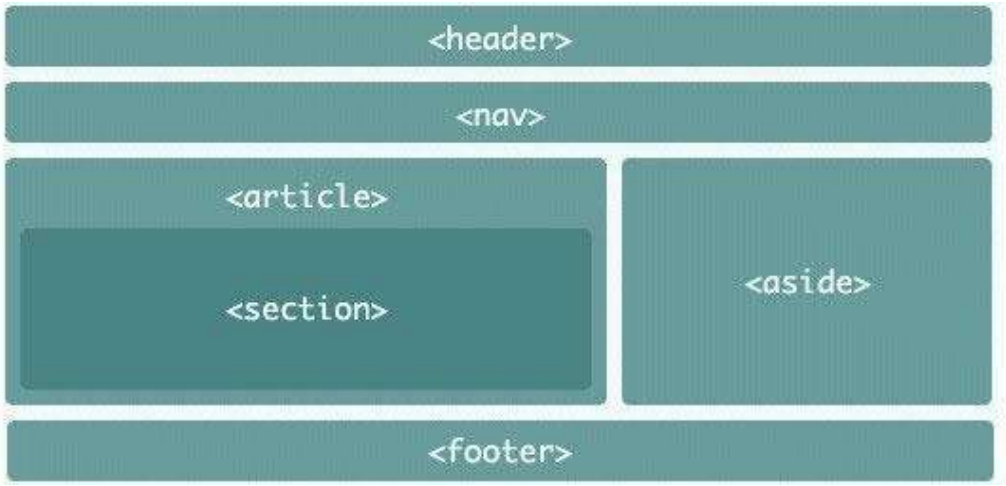
Layout elements

Structural Elements

HTML5 provides us with a variety of tags that we can use when dividing our page into different parts. When browsing the web, you would have noticed that web pages typically have a few common things to them.

For example, a web page will typically have a logo and page navigation area at the top of the page. We would
call this area of the page the header. You may also have noticed that the bottom of t he page may include a list of links and copyright information. We would call this area the footer. The following diagram shows the representation of a few of the main elements of a web
page:



- **header**
- **footer**
- **section**
- **article**
- **nav**
- **aside**
- **div**

Before we begin Day 2 We need to understand some concepts:

- **Ids and Classes and Div**
- **Box Model**
- **Inline and Block elements**
- **Positioning**

IDs and Classes in HTML

The ID is an identifier which must be unique in the whole HTML document. It is used to find an element while linking, scripting, or styling. Whereas, Classes allow CSS and JavaScript to select and access specific elements.

Let us start by making a new file as *idsandclasses.html* and adding an instant boilerplate to get the basic HTML template. Name the <title> tag as **Ids and Classes in HTML** to give the title of the website.

When a new child is born, we urge to give him a name or his identity by which he will be known further. Or if you are having a pet, you must have given him some name to call. In the same way, IDs refer to giving a name to any particular element for its identity. It simply refers giving an identity to an element. We know, no two names can be given to any of the two members of the family. In the same way, one ID can be given to only one element on a website. Therefore, in the below example, the id ***mainBox*** cannot be given to any other element.

Now the question arises what is the need for an ID in HTML? The answer is, while using JavaScript or CSS, we can target one full element and can make the necessary changes in it. In the same way, we can grab the full element and change the border or width or many more things through CSS.

Let us now understand what are classes with an example. Assume that I am having 100 elements in my HTML and I want to give a red background to all the 100 elements. To do this, we have two options. Either we have to select each element and assign a red background to it or we can create a class ***redBG*** and assign a red background to it. Then we can give this class to the elements in which we want a red background color. To avoid confusion, I am assuming that the class redBG is already defined.

One point to note here is we can assign only one ID to a particular element but it is not so in the case of classes. An element can have more than one class in itself. The more classes we add in an element, the more property will get added to it.

Classes are denoted by a dot ‘.’ and ID is denoted by hash ‘#’. For example, to get a redBG class in an element we can simply write that element name followed by .redBG.

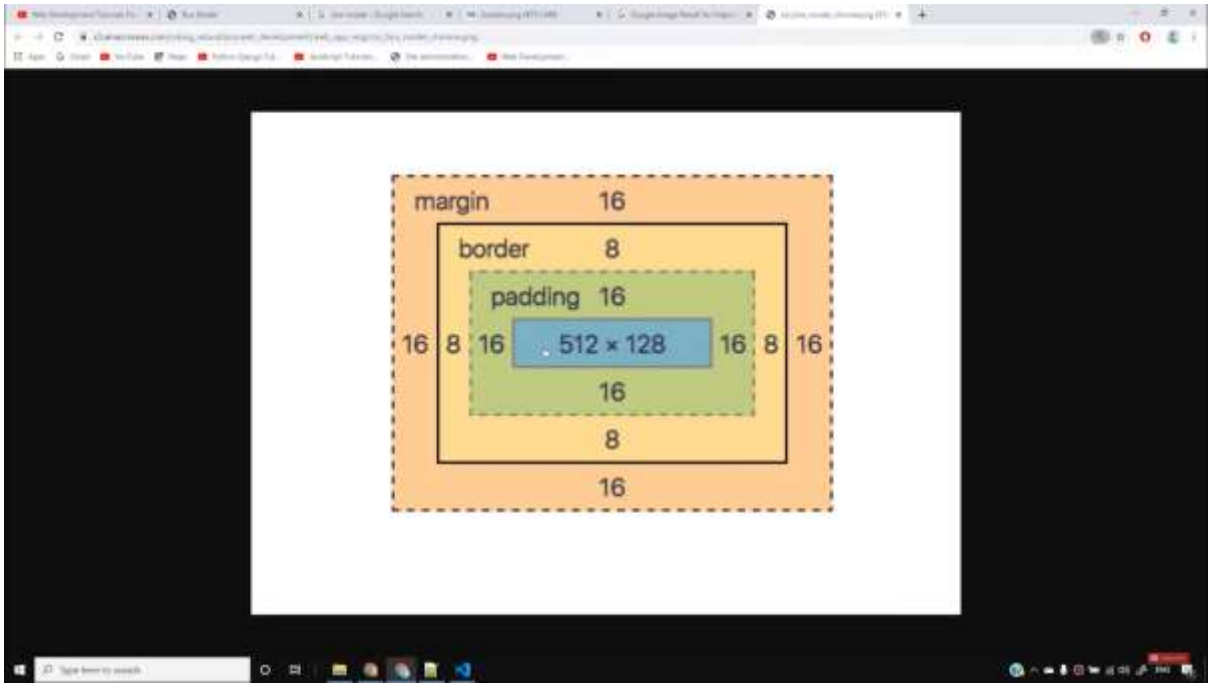
CSS Box Model, Margin and Padding

The box model is a very important topic of CSS and if not understood properly, it can create a lot of confusion in the future. It is the basic framework of web development whether you are making a website using any other language such as Angular or React. The box model helps us to define the **padding**, **border**, and **margin** around an element. So from the below diagram we can see where all these things lie around the element. The element is in the center surrounded by padding, border and margin.

These parts can be explained as-

- **Content-** The content of the box, where text and images appear.
- **Padding-** It clears an area around the content. The padding is transparent.

- **Border-** A border is one that covers the padding and content.
- **Margin-** It clears an area outside the border. The margin is also transparent.



CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe the p resentation of a web page. The language is designed to separate concerns. It a llows the design, layout, and presentation of a web page to be defined separately from content semantics and structure. This separation helps keep source code readable and it is important because a designer can update styles separately from a developer who is creating the page structure or a web editor who is changing content on a page.

A set of CSS rules in a style sheet determines how an HTML document is displayed to the user. It can determine whether elements in the document are rendered at all, whether they appear in some context but not others, how they are laid out on the web page, whether they are rendered in a different order to the order in which they appear within a document, and their aesthetic appearance. We will begin by looking at the syntax of CSS.

Syntax

A CSS declaration is made of two parts:

a property and a value.

The property is the name for some aspect of style you want to change; the value is what you want to set it to.

Here is an example of a CSS declaration:

color: red;

The property is color and the value is red. In CSS, color is the property name f or the foreground color value of an element. That essentially means the color of the text and any text decoration (such as underline or strikethrough). It also sets a currentcolor value. For this declaration to have any effect on an HTML document, it must be applied to one or more elements in the document. We do this with a selector. For example, you can select all the <p> elements in a web page with the p selector. So, if you wanted to make the color of all text in all paragraph elements red, you would use the following CSS ruleset:

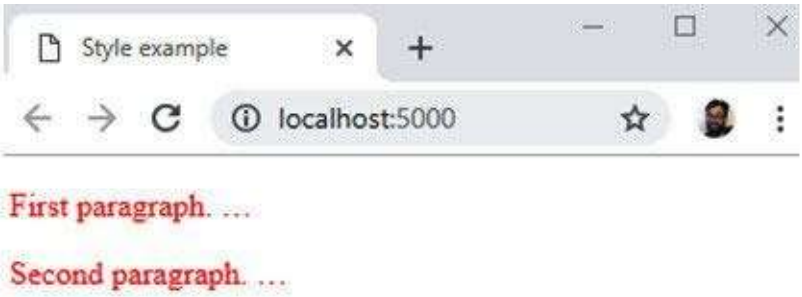
```
p

{

color: red;

}
```

The result of this CSS ruleset applied to an HTML document can be seen in the following figure:



The curly braces represent a declaration block and that means more than one CSS declaration can be added to this block.

Multiple selectors can share a CSS ruleset. We can target these with a comma- separated list. For example, to apply the color red to p elements, h1 elements, and h2 elements, we could use the following ruleset:

```
p, h1, h2 {    color: red;  }
```

Multiple CSS rulesets form a style sheet. The order of these CSS rules in a style sheet is very important as this is partly how the cascade or specificity of a rule is determined. A more specific rule will be ranked higher than a less specific rule and a higher-ranked rule will be the style shown to the end user. We will look at cascade and specificity later in this session:

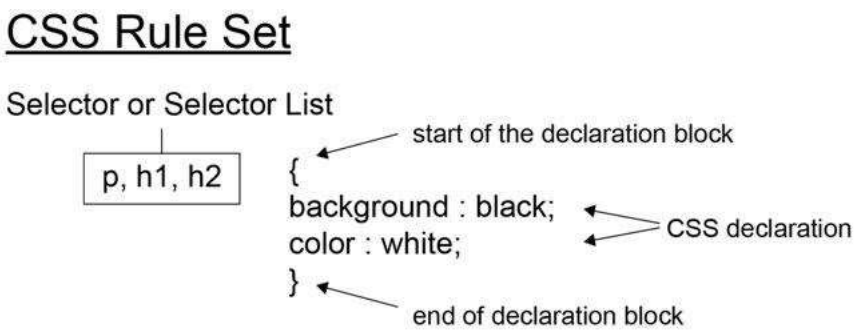


Figure 1.20: A CSS ruleset explained

Element, ID, and Class **Three commonly used selectors are:**

- **Element type: For example, to select all p elements in an HTML document, we use the p selector in a CSS ruleset. Other examples are h 1, ul, and div.**
- **A class attribute: The class selector starts with a dot. For example, given the HTML snippet**
  <h1 class="heading"> Heading</h1>,
  you could target that element with the .heading selector.
- **An id attribute: The id selector starts with a hash symbol. For example, given the HTML snippet**

  ** **
  <!-- login content -->
  ** ,**
  you could target this element with the #login selector.

The Universal Selector (*)

To select all elements throughout an HTML document, you can use the universal selector, which is the asterisk symbol (*). Here is an example snippet of CSS that is often added to web pages; a value is set on the html element and then inherited by all descendant elements:

html

{ box-sizing: border-box; }

Some relative Codes

External StyleSheets

```
<link rel="stylesheet" href="file_path_here" />
```

Selecting a tag and changing the text colour

In CSS we can simply state what type of tag we want to style (such as a p tag below) and then in between the curly brackets, we state what we want to change (below we are changing the text colour of all paragraphs)

```
p {
    color: red;
}
```

Note - you state what property you wish to change (i.e. color) and then follow it with a colon (:).
Then you state what you are changing it to (red) and finally, end the line with a semi-colon (;)

Changing the font

```
font-family: Arial, "Times New Roman";
```

Text Size

```
font-size: 15px;
```

Aligning the text

```
text-align: left;
```

Background colour

```
background: lightblue;
```

Underline Text

```
text-decoration: underline;
```

Height

```
height: 100px;

/*You can also use percentages*/

height: 50%;
```

Width

```
width: 100px;
```

Pseudo Classes

```
a: link
{
    color: black;
}

a: visited
{
    text-decoration: underline;
}

a: hover
{
    background: purple;
}

a:active
{
    color: yellow;
}
```

ID

First we must apply an ID to an element in HTML:

```
<p id="redtext"> Paragraph tag with ID </p>

<p> Paragraph tag without ID</p>
```

Then we can target our ID in CSS using the # symbol:

```
#redtext {
        color: red;
}
```

Classes

Likewise, we must apply classes to our HTML elements:

```
<ul>
        <li>Bread</li>
        <li class="highlight">Cake</li>
        <li>Pasta</li>
        <li class="highlight">Toilet paper</li>
</ul>
```

Then we can target the class in CSS using the . symbol:

```
.highlight {
        background: yellow;
}
```