**What is HTML**

* HTML is the language of the Web. It stands for Hypertext Markup Language.

**What is the purpose of HTML?**

* Its purpose is to let us communicate with the browser the meaning of the content that we wish to place on a web page

**HTML defines a number of tags that we can wrap the contents with.**

* Headings and paragraphs of text
* Images
* Links
* Lists
* Tables
* Forms

**The browser does not display the HTML tags** but uses the tags to interpret the content of the page.

**Tags can encapsulate other tags**. This is called nesting. Nested items are **indented with a tab** to make the document easier to read. The hierarchy that rises from **nesting** is called **DOM** - Document Object Model.

* **What is the DOM?**

The Document Object Model (DOM) is a structured representation of your HTML as generated by the browser, allowing access to the elements of your web page so they may be manipulated

**CSS Review**

**How to add an external CSS :**

<head>

        <meta charset="UTF-8">

        <title>My Travel Blog</title>

        <link rel="stylesheet" href="index.css">

    </head>

**https://www.w3schools.com/cssref/css\_selectors.asp**

### **width | height:**

The width and height properties are used to determine the size of your elements. The values can be expressed in pixels (px) and percentage (%). When working with static content using px is suitable as you will be defining your page to not change. If you are working with responsive design, you will want to use %.

### **overflow:**

This property determines what should happen when the content inside of a container is too much for the container size. You can set the container to hide the additional information that does not fit, show the information no matter what, or have a scroll bar added to the element so the content is contained but still viewable

### **background:**

The background property can modify the background of an element all in one line. This is a shorter version than splitting each property on its own line of code. The color can be defined using hex, rgb or semantic code.

p {

background: #ffffff url("cherries.png") no-repeat fixed center;

}

### **border:**

This property adjusts all border elements in one line. The first value is the border thickness. The second value is the border type. The third value is the border color. The color can be hex, rgb or semantic code.

The border property can also be broken down into separate lines using border-width, border-style and border color. Additionally, you can select very specifically which border you want to style by using border-top, border-bottom, border-right, border-left.

### **border-radius**

This property allows the corners of your border to be given a rounded appearance. The values can be set using px or %. Given appearance on the border

border-radius: 10px;

### **text-align:**

This property is used to set the horizontal alignment of any text. Text can be centered, or aligned to the left or right, or justified. This property will only work if the property is also display block.

## What is a span tag?

If you wanted to highlight specific text within a paragraph or division, <span> can be used.  Unlike <div> which puts the next <div> on a separate line, <span> by default puts the element on the same line.

<h1>Hello <span>World</span> </h1>

**Inline-block:**

width: 200px;

        margin:15px;

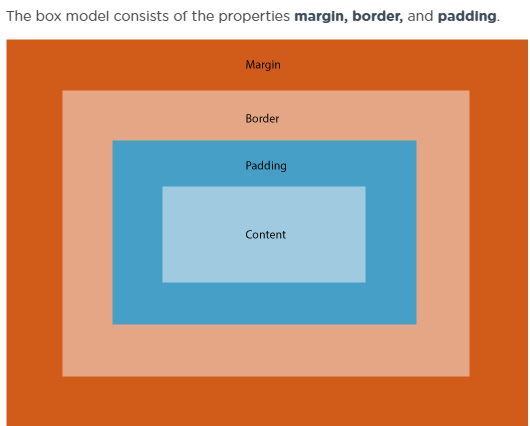
        padding:20px;

        font-size: 14px;

        background: #eee;

        display: inline-block;

# **Box Model - Margin, Border, Padding**



The colspan attribute in HTML specifies **the number of columns a cell should span**. It allows the single table cell to span the width of more than one cell or column. It provides the same functionality as “merge cell” in a spreadsheet program like Excel

**colspan=”2” specify the width**

The **rowspan** attribute in HTML specifies **the number of rows a cell should span**. That is if a row spans two rows, it means it will take up the space of two rows in that table. It allows the single table cell to span the height of more than one cell or row.

**rowspan=”2” specify the width**

<form action="/example.html" method="POST">  
</form>

In the above example, we’ve created the skeleton for a <form> that will send information to **example.html** as a POST request:

* The action attribute determines where the information is sent.
* The method attribute is assigned a HTTP verb that is included in the HTTP request.

**Input Numbers**

<form>  
  <label for="volume"> Volume Control</label>  
  <input id="volume" name="volume" type="range" min="0" max="100" step="1">  
</form>

**The <datalist>** is used with an <input type="text"> element. The <input> creates a text field that users can type into and filter options from the <datalist>. Let’s go over a concrete example:

**Checking Text Length**

In the previous exercise, we were able to use min and max to set acceptable minimum and

<form action="/example.html" method="POST">  
  <label for="summary">Summarize your feelings in less than 250 characters</label>  
  <input id="summary" name="summary" type="text" minlength="5" maxlength="250" required>  
  <input type="submit" value="Submit">  
</form>

**Matching a Pattern**

In addition to checking the length of a text, we could also add a validation to check how the text was provided. For cases when we want user input to follow specific guidelines, we use the pattern attribute and assign it a *regular expression*, or regex. Regular expressions are a sequence of characters that make up a search pattern. If the input matches the regex, the form can be submitted.

Let’s say we wanted to check for a valid credit card number (a 14 to 16 digit number). We could use the regex: [0-9]{14,16} which checks that the user provided only numbers and that they entered at least 14 digits and at most 16 digits

<form action="/example.html" method="POST">  
  <label for="payment">Credit Card Number (no spaces):</label>  
  <br>  
  <input id="payment" name="payment" type="text" required pattern="[0-9]{14,16}">  
  <input type="submit" value="Submit">  
</form>