



HTML  
<hyper text markup language>

&

CSS  
Cascading Style Sheets



# HTML

**HTML** stands for Hyper Text Markup Language, it is easy and fun to learn. It describes the structure of web pages. HTML5 is the fifth and current major version of the HTML standard.

HTML (**H**ypertext **M**arkup **L**anguage) uses a markup system composed of elements which represent specific content.

Markup means that with HTML you declare what is presented to a viewer, not how it is presented. Visual representations are define by Cascading Style Sheet (CSS) and realized by browsers.

HTML is sometimes called a programming language but it has no logic, so is a **markup language**. HTML tags provide semantic meaning and machine-readability to the content in the page.

An element usually consists of an opening tag (<element\_name>), a closing tag (</element\_name>), which contain the element's name surrounded by angle brackets, and the content in between:

**<element\_name>content</element\_name>**

There are some HTML elements that don't have a closing tag or any contents. These are called void elements.

Void elements including **<img>**, **<meta>**, **<link>** and **<input>**.

Element names can be thought of as descriptive keywords for the content they contain, such as video, audio, table, footer.

A HTML page may consist of potentially hundreds of elements which are then read by a web browser, interpreted and rendered into human readable or audible content on the screen.

For this, it is important to note the difference between elements and tags:

**Elements:** video, audio, table, footer

**Tags:** **<video>**, **<audio>**, **<table>**, **<footer>**, **</html>**, **</body>**

# HTML TAGS

HTML Tags are element names surrounded by angle brackets.

In HTML we start and end tags. Look at the example.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5 </head>
6 <body>
7   <p>This is the Content Space where we are writing
   content about Jetking Blockchain</p>
8 </body>
9 </html>
```

- <!DOCTYPE html>: This declares the document type in which is HTML5.
- <html>: This element encloses inside of an html document; It includes tags, elements, style sheets, scripts, text, multimedia and a lot more.
- <head>: This element encloses the metadata of a document which will not be displayed on the main content of a web page; This could

include style sheets, scripts, <title>, <meta> tags and a lot more.

- <title>: This element defines the title of a web page; It appears on the upper-part of a browser.
- <body>: This element encloses elements like <h1>, <p>, <img>, <b>, <i> and a lot more.
- <p>: This element defines a paragraph.

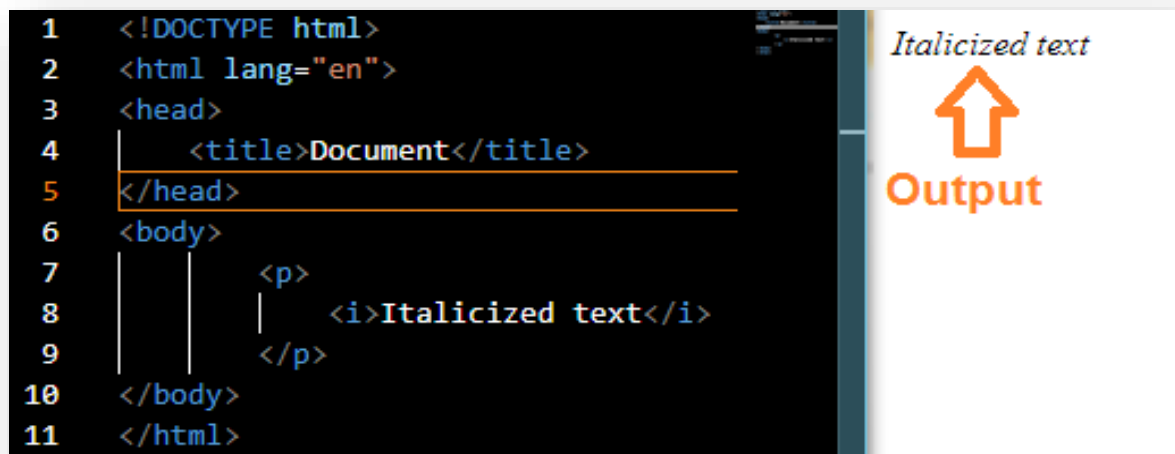
## HTML ELEMENTS

An HTML elements is usually composed of a “Start Tag”, “Element Content” and an “End Tag”.

<Number1> 10 </Number>

Number1 = 10

Example:



Empty Elements are elements that do not have an element content and an end tag.

A list of commonly used Empty Elements:

- `<meta />`: The meta element represents metadata that cannot be represented by other HTML meta-related elements like `<link>`, `<script>`, `<script>`, `<title>`.
- `<link />`: The link element specifies relationships between the current document and an external resource. This element is most commonly used to link to style sheets.
- `<img />`: The img tag element embeds an image into the document.
- `<br />`: The br element produces a line break in text. It is useful for writing a poem or an address, where the division of lines is significant.
- `<hr />`: The hr element represents a thematic break between paragraph-level element.
- `<input />`: the input element is used to create interactive controls for web-based forms in order to accept data from the user.

# HTML ATTRIBUTES

HTML attributes are used to add more information to an HTML Element.

## **Important Things to remember:**

- HTML attributes are found in HTML tags.
- HTML attributes only appear at start tags. It will never be on end tags.
- HTML attributes can have multiple attributes.
- HTML attributes are composed of name/value pairs.
- There are some attributes that can be user on all HTML Elements though they may not have effects on some elements. They are called as Global Attributes.

An HTML Attribute is composed of:

- An attribute name
- An equal sign =.
- A value surrounded by quotation marks “value”.

i.e. Student\_name = “Tushar”

You can also use single quotation marks depending on the situation.

### Attribute lang Example:

```
1 <!DOCTYPE html>  
2 <html lang="en">
```

We use the lang attribute to define the language of an HTML file.

The language defined above is American English.

### Attribute href Example:

```
<a href="https://jetking.com/" title="Official Website Link">Official Website Link</a>
```

The title attribute provide a tool tip for HTML Elements.

Unfortunately, it doesn't work on mobile devices.

### Attribute style Example:

```
<p style ="font-size: 40px; color: ■ gold;"><i>This is the Content Space where we  
are writing content about Jetking Blockchain</i></p>
```

We can use the style attribute to change its font-size and color.



## Attributes id and class Example:

```
<h1 id="Branches">Branches of Jetking Infotrain Limited</h1>  
<h1 class="Chandigarh_Branch">SCO 96-97, Sector 34-A, Chandigarh</h1>  
<h1 class="Delhi_Branch">Laxmi Nagar Branch, Delhi</h1>
```

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page.

The id and class attributes give reference to elements inside an HTML document.

Multiple elements can have the same class value/names.

The id's value should be unique for each element.

## HTML COMMENTS

HTML Comments are text, phrases or sentences inside an HTML file.

They are only show in codes and not rendered by a browser.

HTML Comments help both beginners and experience web developers to early organize their codes.

They act like sticky notes in HTML files.

An HTML Comment starts with

<!--and ends with-->

```
For single line comments
<!--This are the comments of HTML file-->

For multi-line comments
<!--These are Multiline Comments Written in HTML-->
```

## HTML SECTIONS

HTML Sections groups different HTML Element (ex: heading, paragraph, images etc.) together.

Creating sections is essential for organizing and styling web contents.

- <div>: It is used for grouping a large group of HTML elements like navigation, header, main content, footer, images; it is a block-level element.

- `<span>`: It is used for grouping text into a paragraph and very few HTML elements. It is an inline element.

Here is a list of HTML5 Semantic Elements that work similarly with `<div>` element:

- `<nav>`: defines a navigation list or a bar.
- `<header>`: defines a header.
- `<main>`: defines the main content.
- `<footer>`: defines a footer.

### Example of `<div>`:

```
<div style="background: ■purple; width: 400px; height: 400px;">
  <p>This is a paragraph i am writing for explaining the HTML5 Sections</p>
</div>
<div style="background: ■yellow; width: 400px; height: 400px;">
  <p>This is a paragraph i am writing for explaining the HTML5 Sections</p>
</div>
```

### Output:



Here as you can see the two `<div>` elements are separated from each other they also have different style, That's one of the advantages of HTML sections, it lets style specific parts of an HTML document.

## Example of <span>:

```
<p>  
<span style="font-size: 30px"> </span>  
<span style="color: blue">am c color with name blue</span>  
<span style="color: red"> staying with another color red</span>  
<span style="color: brown"> tThis is written for understanding span element</span>  
</p>
```

am a color with name blue staying with another color red This is written for understanding span element

## Output:

Here as you can see the <span> elements did not separate the paragraph into multiple lines.

## Important Note:

The <span> element is good for styling and organizing few and inline elements while the <div> element is good for only larger elements.

# HTML HEADINGS

HTML Headings usually contain a title or a main topic of certain content

HTML Headings are Block-level elements.



## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7
8     <h1>Heading 1</h1>
9     <h2>Heading 2</h2>
10    <h3>Heading 3</h3>
11    <h4>Heading 4</h4>
12    <h5>Heading 5</h5>
13    <h6>Heading 6</h6>
14 </body>
15 </html>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Here you need to remember that the lower the number the longer the font-size.

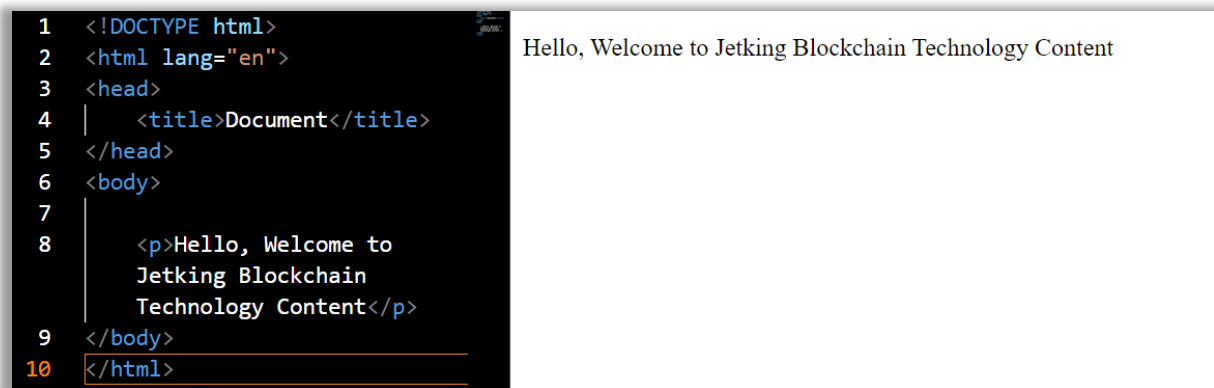
## HTML PARAGRAPH

HTML Paragraph are mainly used to group sentence.

The <p> element defines HTML Paragraphs.

HTML Paragraph are Block-level elements.

## Example:



# HTML LINKS

HTML Links are very necessary for all web pages. All web sites have links. HTML Links are hyperlinks.

Links help users jump from their current web page location to another.

To make HTML Links we need use <a> element with the href attribute that specified URL to a page.

Start tag: <a>

Href attribute: href="index.html"

Element content: A text Home page./A Image.

End tag: <a>.

# LINE BREAKES AND HORIZONTAL RULE

An HTML Line Break produces a line break in a text. It is an empty element.

The `<br>` element defines an HTML Line Break.

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7 |
8 |   I am the first line.<br>
9 |   I am the second line. <br>
10 |  I am the third line. <br>
11 </body>
12 </html>
```

I am the first line.  
I am the second line.  
I am the third line.

HTML Horizontal Rule renders a line in web browser. It is an empty element.

The `<hr>` element defines an HTML Horizontal Rule.

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7 |
8 |   <p>This is First paragraph</p>
9 |   <hr>
10 |  <p>This is Second Paragraph</p>
11 </body>
12 </html>
```

This is First paragraph

---

This is Second Paragraph

# HTML5 TEXT FORMATTING

Formatting text is very essential for website.

One of its uses is to help readers know the important parts of a web content easily.

All elements listed below are inline elements.

- **<b>**: bolds a text.
- **<i>**: italicize a text.
- **<u>**: underline a text.
- **<code>**: define a text as a code.
- **<tt>**: typewriter text.
- **<small>**: makes a text smaller.
- **<em>**: emphasizes a text, may be boldened depending on the browser.
- **<mark>**: marks a text like a highlighting pen.
- **<q>**: enquotes a text.
- **<s>**: strikes through a text.



## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7   <b>Bolds a text</b>
8   <br>
9   <i> italicize a text.</i>
10  <br>
11  <u>underline a text.</u>
12  <br>
13  <code>define a text as a
14  code</code>
15  <br>
16  <tt>typewriter text</tt>
17  <br>
18  <small>makes a text small</
19  small>
20  <br>
21  <em>emphasizes a text</em>
22  <br>
23  <mark>marks a text</mark>
24  <br>
25  <q>enquotes a text</q>
26  <br>
27  <s>strikes through a text</s>
28 </body>
29 </html>
```

**Bolds a text**

*italicize a text.*

underline a text.

define a text as a code

typewriter text

makes a text small

*emphasizes a text*

**marks a text**

“enquotes a text”

~~strikes through a text~~

# BLOCK-LEVEL & INLINE ELEMENTS

An HTML Block-level Element is an element that creates a horizontal boundary block.

It always start on a new line and takes the full width available.

## **Block-Level Elements are:**

- `<h1>` - `<h6>`.
- `<p>` element.
- `<div>` element.

An Inline Element is an element that does not create its own horizontal boundary block.

It does not start on a new line.

## **Inline Elements are:**

- `<span>` element.
- `<a>` element.
- `<img>` element.

# HTML5 IMAGES

HTML Images are indeed needed or required for any website.

Images help a web site become more attractive for visitors.

Images a web page without even a single image would you still browse it? Of course nor.

To put an image on our website we simply need to use `<img>` element with an `src` attribute to define the URL or the location of an image.

The <img> is an empty and inline element.

### **Attribute src:**

We use src attribute to specify an image's URL or file path.

### **Attribute alt:**

Sometimes images may not load on the user's browser because of slow internet connection, slow server, image is deleted from directory or wrong URL Value is specified in the src attributes.

The alt attribute provides an alternate text for an image.

### **Image Sizing (Width & Height):**

To resize an image we just need to use width attribute to change its width and the height attribute to change its height.

The value is typically in pixels.

### **Floating Image (left or right):**

We can float an image to left or right side of a text.

To achieve that we need to use style attribute.

With the float CSS property, along with left or right value.

## Image as a Link:

There are some situations that we need an image to act as a link.

To do that just enclose the image with an `<a>` element with its href attribute.

## Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML5 Images Example Script</title>
</head>
<body>
  <a href="https://jetking.com/">  </a>
  
  <p>
    <a href="https://jetking.com/">
      
    </a>
  </p>
  <p>
    The image is floated at left side </p>
  </p>
  The image is floated at right side </p>
  
  
  
</body>
</html>
```



# HTML5 TABLES

HTML Tables represents data in a tabular form. It is composed of columns and rows of s=cells that contain data.

The `<table>` element defines an HTML Table.

## **HTML Table Elements:**

- `<thead>`: defines a table header.
- `<tbody>`: defines a table body.
- `<tfoot>`: defines a table footer.
- `<tr>`: defines a table row.
- Colspan attribute: defines how many columns should a cell span.
- `<th>`: defines a table heading.
- `<td>`: defines a table/cell.
- Rowspan attribute: defines how many rows should a cell span.
- `<colgroup>`: specifies column properties for each column in a colgroup.
- `<caption>`: defines a caption for a table.

## Example:

<pre>1 &lt;!DOCTYPE html&gt; 2 &lt;html lang="en"&gt; 3 &lt;head&gt; 4   &lt;title&gt;Document&lt;/title&gt; 5 &lt;/head&gt; 6 &lt;body&gt; 7   &lt;table&gt; 8     &lt;thead&gt; 9       &lt;tr&gt; 10         &lt;th&gt; 11           First Name 12         &lt;/th&gt; 13         &lt;th&gt; 14           Last Name 15         &lt;/th&gt; 16       &lt;/tr&gt; 17     &lt;/thead&gt; 18     &lt;tbody&gt; 19       &lt;tr&gt; 20         &lt;td&gt; 21           Babu 22         &lt;/td&gt; 23         &lt;td&gt; 24           Sharma 25         &lt;/td&gt; 26       &lt;/tr&gt; 27     &lt;/tbody&gt; 28     &lt;tfoot&gt; 29       &lt;tr&gt; 30         &lt;td&gt;Footer&lt;/td&gt; 31         &lt;td&gt;Footer&lt;/td&gt; 32       &lt;/tr&gt; 33     &lt;/tfoot&gt; 34   &lt;/table&gt; 35 &lt;/body&gt; 36 &lt;/html&gt;</pre>	<table><tr><th>First Name</th><th>Last Name</th></tr><tr><td>Babu</td><td>Sharma</td></tr><tr><td>Footer</td><td>Footer</td></tr></table>	First Name	Last Name	Babu	Sharma	Footer	Footer
First Name	Last Name						
Babu	Sharma						
Footer	Footer						

## HTML Table Attributes:

- Align: defines the alignment of a table.
- Bgcolor: defines the background color of a table.
- Border: defines the size of the frame surrounding a table.
- Cellpadding: defines the space between the content of a cell and its border.

- Cellspacing: defines the space between two cells.
- Frame: defines which side of the frame surrounding the must be displayed.
- Rules: defines where rules should appear in a table.
- Summary: defines an alternative text that summarizer the content of the table.
- Width: defines the width of a table.

## Example:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Document</title>
5  </head>
6  <body>
7  |   <table style="background-color: ■ silver; width: 80%; margin: 0 auto; border: 2px solid □ black">
8  |   |   <colgroup>
9  |   |   |   <col style="background-color: ■ blue">
10 |   |   |   <col style="background-color: ■ red">
11 |   |   |   <col style="background-color: ■ green">
12 |   |   </colgroup>
13 |   |   <thead>
14 |   |   |   <tr>
15 |   |   |   |   <td>First Name</td>
16 |   |   |   |   <td>Last Name</td>
17 |   |   |   |   <td>Nationality</td>
18 |   |   |   </tr>
19 |   |   </thead>
20 |   |   <tbody>
21 |   |   |   <td>Babu</td>
22 |   |   |   <td>Sharma</td>
23 |   |   |   <td>Indian</td>
24 |   |   </tbody>
25 |   |   <tbody>
26 |   |   |   <td>Tushar</td>
27 |   |   |   <td>Devwal</td>
28 |   |   |   <td>Indian</td>
29 |   |   </tbody>
30 |   |   <tbody>
31 |   |   |   <td>Mohit</td>
32 |   |   |   <td>Kumar</td>
33 |   |   |   <td>Indian</td>
34 |   |   </tbody>
35 |   </table>
36 </body>
37 </html>
```

## Output:

First Name	Last Name	Nationality
Babu	Sharma	Indian
Tushar	Devwal	Indian
Mohit	Kumar	Indian

# HTML5 LISTS

The Elements represent HTML Lists that can be used to group a collection of items with and/or without order.

- `<ul>`: defines an unordered list where the order is meaningless and it typically bulleted.
- `<ol>`: defines an ordered list where the order is meaningful and its typically numbered.
- `<li>`: a child element of both `<ul>` and `<ol>` elements that defines a list item.



## Example:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Document</title>
5  </head>
6  <body>
7  |   <p>Fruits</p>
8  |   <ul>
9  |       <li>Apple</li>
10 |       <li>Oranges</li>
11 |       <li>Grapes</li>
12 |   </ul>
13
14 |   <p>Alphabets</p>
15 |   <ol>
16 |       <li>A</li>
17 |       <li>B</li>
18 |       <li>C</li>
19 |       <li>D</li>
20 |   </ol>
21 </body>
22 </html>
```

### Fruits

- Apple
- Oranges
- Grapes

### Alphabets

1. A
2. B
3. C
4. D

## Nested HTML Lists Example:

Sometimes we have to nest list to make the data represent easier to understand.

### Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7 |   <ul>
8 |     <li>General Data</li>
9 |     <ul>
10 |       <li>Specific Data</li>
11 |       <li>Specific Data</li>
12 |       <li>Specific Data</li>
13 |     </ul>
14 |     <li> General Data</li>
15 |     <ol>
16 |       <li>Specific Data</li>
17 |       <li>Specific Data</li>
18 |       <li>Specific Data</li>
19 |     </ol>
20 |   </ul>
21 </body>
22 </html>
```

- General Data
  - Specific Data
  - Specific Data
  - Specific Data
- General Data
  1. Specific Data
  2. Specific Data
  3. Specific Data

## HTML5 FORMS

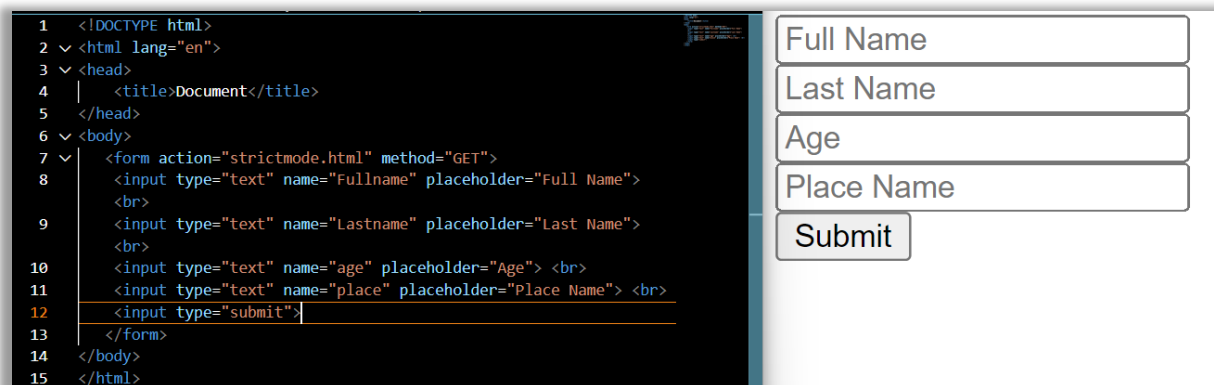
HTML forms can be used to collect user data, make a login form, registration form, a search form, and a contact form.

The <form> elements defines an HTML form.

## HTML Form Attributes:

- **Accept-charset:** It defines the character set of the form.
- **Action:** Specifies the form-handler/page-handler page.
- **Autocomplete:** It tells the browser to automatically complete the form elements.
- **Enctype:** It defines the encoding data submitted by the form.
- **Method:** It defines the HTTP method to use when submitting a form. e.g.: GET, POST, HEAD.
- **Name, Id, Class:** It is used for identifying the form.
- **Novalidate:** It is telling the browser not to validate the form.
- **onSubmit:** It is used to call a JavaScript function. Like the form is going to validate when the form is submitted.
- **Target:** It defines the target of the form e.g. target = “\_blank”. By giving this we can open the form in a new tab.

## Example:



The image shows a side-by-side comparison of an HTML form and its source code. On the left, the HTML code is displayed in a dark-themed editor with line numbers 1 through 15. It defines a form with the action 'strictmode.html' and method 'GET'. The form contains four text input fields with names 'Fullname', 'Lastname', 'age', and 'place', each with a corresponding placeholder. A submit button is also present. On the right, the rendered form is shown as a light gray box with rounded corners. It contains four stacked text input fields with the placeholders 'Full Name', 'Last Name', 'Age', and 'Place Name', followed by a 'Submit' button.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5 </head>
6 <body>
7   <form action="strictmode.html" method="GET">
8     <input type="text" name="Fullname" placeholder="Full Name">
9     <br>
10    <input type="text" name="Lastname" placeholder="Last Name">
11    <br>
12    <input type="text" name="age" placeholder="Age"> <br>
13    <input type="text" name="place" placeholder="Place Name"> <br>
14    <input type="submit">
15  </form>
</body>
</html>
```

## Attribute Method:

The method attribute specifies which HTTP method to use when submitting input data to the form-handler page.

### GET v/s POST Methods

- GET Method is going to give the default value.
- It is less secure.
- Data is written on the URL address.
- It is very likely to hack the information.
- Don't submit the personal information like passwords, account number etc.

Post method is not going to give the default value.

- It is more secure.
- Data is not written on the URL Address.
- It is very difficult to Hack the information.

- Is you want to Submit the personal information we can use it.

## **HTML Form Labels:**

An HTML Form Lable is an inline element.

It can be associated with form element or form controls like an input an input, text area and select for accessibility purpose.

The `<label>` element defines as HTML form label.

To associate a label to an input we can use “for” attribute with the value same as the input’s “id” attribute.

# CSS3 INTRODUCTION

CSS stands for Cascading Style sheet. It is a scripting language used for describing the presentation of an HTML document.

CSS mainly explains how the elements should be presented on screen.

We use CSS for defining styles for web pages including design, layout and variations in display across various devices and screen sizes.

In the process of constructing a house, we are using cement, bricks, stones, and sand for laying basement and raising the walls.

The process of using bricks and cement is related to using HTML5 for designing a website.

In HTML the bricks are nothing but the HTML Tags.

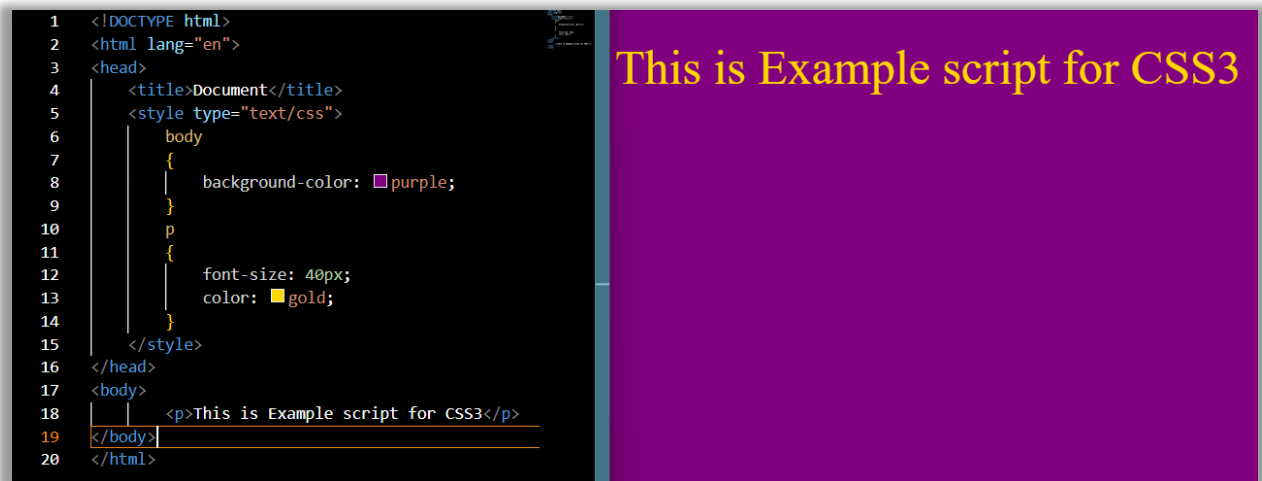
After completion of basement, pillars, floors and walls, RCC roofs we will start painting the house.

The process of painting the house is related to using CSS in designing a website.

After completion of construction and painting, we will start applying for electricity from electricity board.

The process of applying electricity connection to you newly constructed house is related to using JavaScript.

### Example:





# CSS SYNTAX

A class Syntax is composed of a selector and a declaration block.

## **Example:**

```
div
{
background-color: purple;
width: 200px;
height: 200px;
}
```

## **Div:**

The div is a selector or a group of selectors target the elements that we want to style the web page.

## **Declaration Blocks:**

The declaration blocks are grouped in Blocks. Each declaration block contains CSS declarations which are wrapped by an opening curly brace { and closing curly brace }.

## **CSS Declarations:**

The CSS declaration are contained inside a declaration block. Each declaration has to be separated by a semicolon ;.

It is composed of a pair of a property and a value separated by a colon :.

Here the property refers to human-readable identifier that indicates which stylistic feature to change.

And the value refers to the property which indicates how the specified feature should be changed.

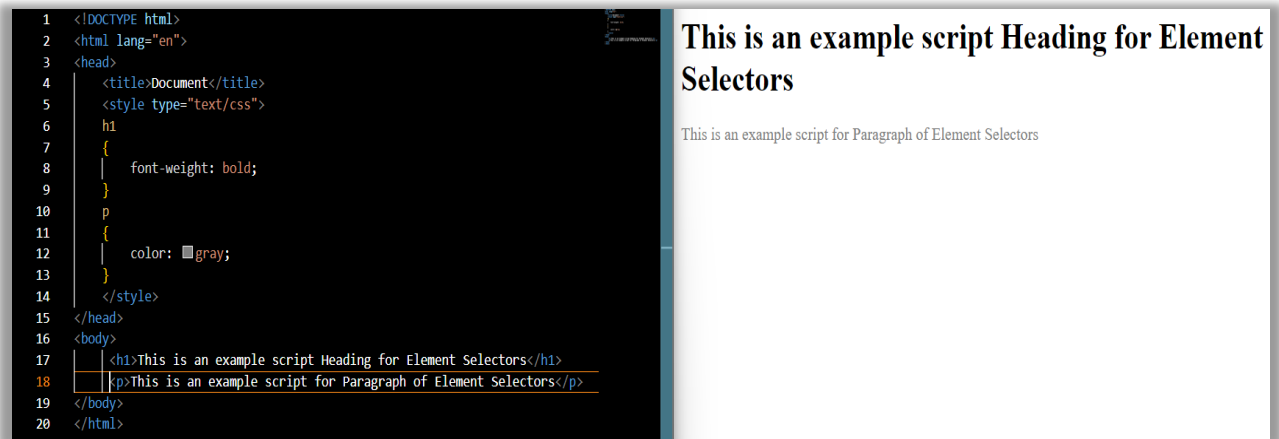
## **CSS SELECTORS**

In CSS3, there are four different types of CSS selectors.

1. Element Selectors
2. Class Selectors
3. ID Selectors
4. Grouping Selectors

## Element Selectors:

Element Selectors are the simplest way to target all the elements of given type.



The screenshot shows an HTML document with the following code:

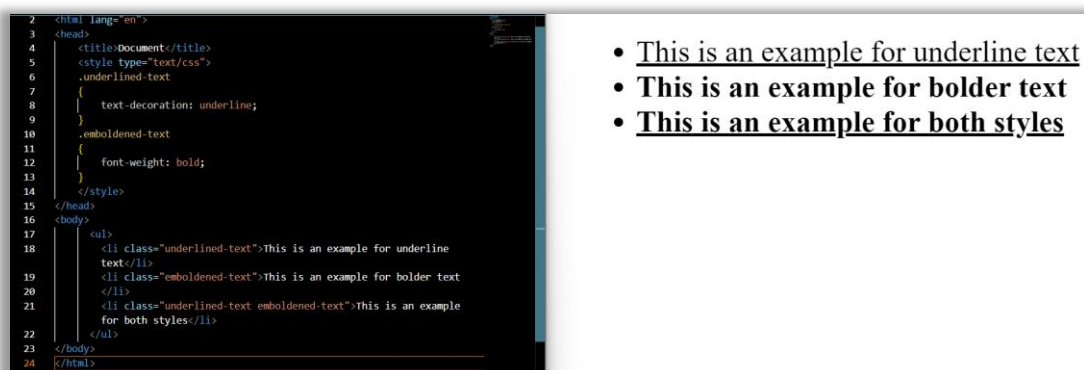
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     h1
7     {
8       font-weight: bold;
9     }
10    p
11    {
12      color: gray;
13    }
14  </style>
15 </head>
16 <body>
17   <h1>This is an example script Heading for Element Selectors</h1>
18   <p>This is an example script for Paragraph of Element Selectors</p>
19 </body>
20 </html>
```

The rendered output on the right shows a bold heading "This is an example script Heading for Element Selectors" and a gray paragraph "This is an example script for Paragraph of Element Selectors".

## Class Selectors:

Multiple elements can have the same class values. CSS class selector are very useful when targeting different elements. The class selector consists of a dot (.) followed by a class name. A class name is defined using the class attribute.

An HTML Element can have multiple class names separated by a white space.



The screenshot shows an HTML document with the following code:

```
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     .underlined-text
7     {
8       text-decoration: underline;
9     }
10    .emboldened-text
11    {
12      font-weight: bold;
13    }
14  </style>
15 </head>
16 <body>
17   <ul>
18     <li class="underlined-text">This is an example for underline
19     text</li>
20     <li class="emboldened-text">This is an example for bolder text
21     </li>
22     <li class="underlined-text emboldened-text">This is an example
23     for both styles</li>
24   </ul>
25 </body>
26 </html>
```

The rendered output on the right shows three list items:

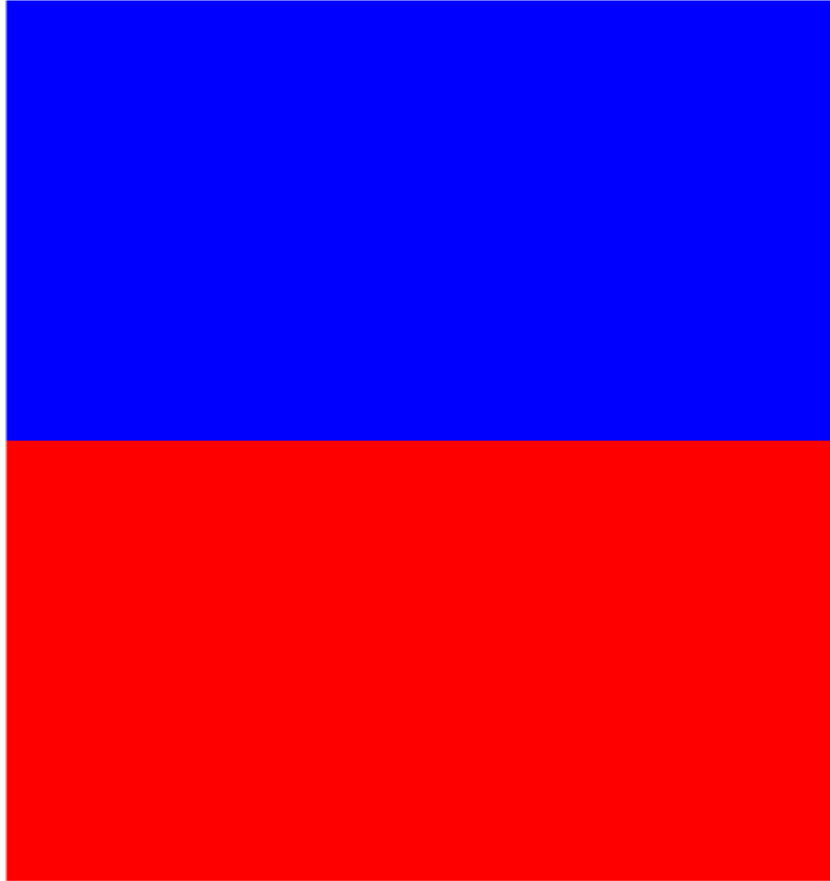
- This is an example for underline text
- This is an example for bolder text**
- This is an example for both styles**

## ID Selectors:

The ID Selector is very useful when targeting a single element. An ID selector consists of Hash (#) followed by an ID name. An ID name is defined using the id attribute.

We can target only specific HTML elements with the given class or id.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5      <style type="text/css">
6          #blue
7          {
8              background-color: ■blue;
9              width: 200px;
10             height: 200px;
11         }
12         #red
13         {
14             background-color: ■red;
15             width: 200px;
16             height: 200px;
17         }
18         p.red
19         {
20             color: ■red;
21         }
22         p#green
23         {
24             color: ■green;
25         }
26     </style>
27 </head>
28 <body>
29     <div id="blue"></div>
30     <div id="red"></div>
31     <p class="red">I am a paragraph with class name red</p>
32     <p id="green">i am a paragraph with id name green</p>
33 </body>
34 </html>
```



I am a paragraph with class name red  
i am a paragraph with id name green

## Grouping Selectors:

A grouping selectors consists of selectors separated by commas.

```
<style type="text/css">
  h1,p
  {
    font-family: 'Times New Roman', Times, serif;
    color: purple;
  }
</style>
</head>
<body>
  <h1>This is an Heading</h1>
  <p>This is an Paragraph</p>
</body>
</html>
```

This is an Heading

This is an Paragraph

# CSS COMMENTS

CSS Comments can be used to explain CSS codes and make it more readable. CSS comments are very useful for people who easily forgot the purpose of the script or things used in the script.

But these comments are not displayed on the browsers.

A typical CSS comment start with `/*` and ends with `*/`.

## Example:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5      <style type="text/css">
6          p
7          {
8              color: purple;
9              /* This is a single line comment */
10         }
11         /* This is multi-line comments which is
12         not
13         displayed on the browser when there are
14         executed*/
15     </style>
16 </head>
17 <body>
18     <p>This is a single line comment</p>
19     <p>This is starting of multi-line comments</p>
20     <p>This is ending of multi-line comments</p>
21 </body>
22 </html>
```

This is a single line comment

This is starting of multi-line comments

This is ending of multi-line comments

# CSS INSERTING WAYS

In CSS3 Inserting, we are having mainly three simple ways to insert CSS into a HTML document.

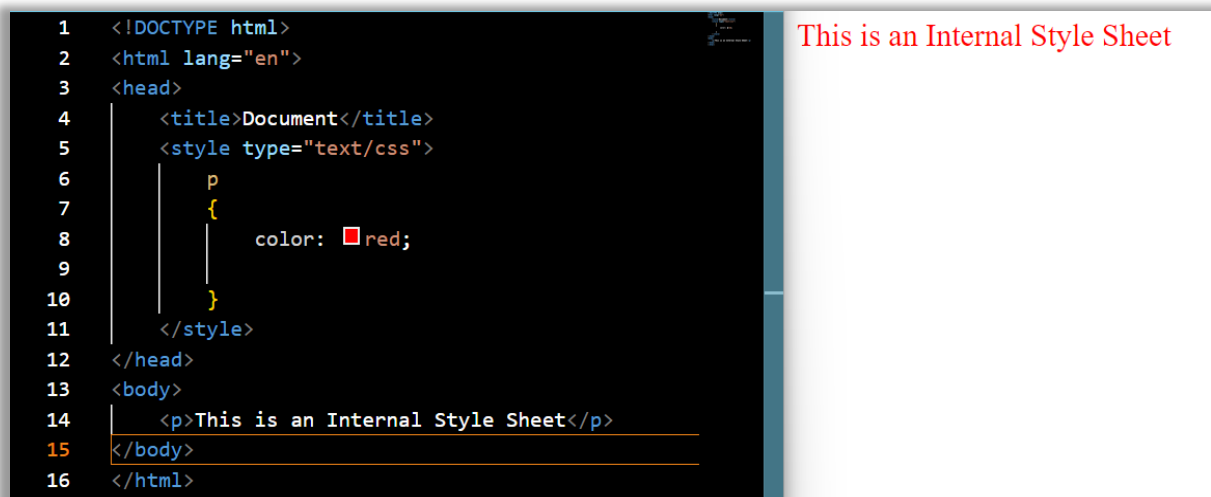
1. Internal Style Sheet.
2. External Style Sheet.
3. Inline Style Sheet.

## Internal Style Sheet:

An internal style sheet is useful when a single HTML document has its unique style. With this, CSS syntax should be included in the `<style>` element. The `<style>` element should only be enclosed inside the `<head>` element.

The “text/css” value indicates that the content is CSS.

## Example:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     p
7     {
8       color: red;
9     }
10  </style>
11 </head>
12 <body>
13   <p>This is an Internal Style Sheet</p>
14 </body>
15 </html>
```

This is an Internal Style Sheet



## External Style Sheet:

An External Style Sheet can be very useful when multiple HTML pages have the same style. This can be achieved by having an external style sheet file.

An external style sheet file must be saved with a .css file extension like “style.css”. then we need to use <link> element to include that file into our HTML page.

There should be no <style> element inside a CSS file.

### Example:

First we need to create one style.css file for linking the CSS into our HTML web page.

The style.css file contains the following codes.

```
1  body
2  {
3  |    background-color: ■ brown;
4  |  }
5  p
6  {
7  |    color: ■ white;
8  |  }
```

Then write the script for HTML file for linking the above CSS file called “style.css”.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 |   <link rel="stylesheet" href="style.css">
6 </head>
7 <body>
8 |   <p>This is an External Style Sheet</p>
9 </body>
10 </html>
```

This is an External Style Sheet

## Inline Styling:

Inline Styling is useful styling a single element. It is done by using the style attribute. Its value contains CSS declaration.

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7 |   <p style="font-size: 100px;
8 |     color: purple">This is an Inline Style
9 |     Sheet</p>
10 </body>
11 </html>
```

This is an Inline  
Style Sheet

# CSS COLORS

CSS Colors are commonly specified using the following:

1. Color Name.
2. Hexadecimal Color Value.
3. RGB Color Value.

## Color Name:

It defines any valid color by its color name e.g. Black, White etc.

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     #color-name
7     {
8       color: purple;
9     }
10  </style>
11 </head>
12 <body>
13   <p id="color-name">This is an color name
14   in CSS</p>
15 </body>
16 </html>
```

This is an color name in CSS

## Hexadecimal Color Value:

Hexadecimal Color value can also called as Hex color value. It defines any valid color by a hash

followed by letters and/or numbers e.g.  
#fffff, #00000.

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     #hex-color
7     {
8       color: □ #000000;
9     }
10  </style>
11 </head>
12 <body>
13   <p id="hex-color">This is an Hexadecimal
14     color in CSS</p>
15 </body>
16 </html>
```

This is an Hexadecimal color in CSS

## RGB color Value:

RGB stands for Red, Green and Blue. It defines any valid color using the rgb() CSS functions by this,

Format: rgb(<number>,<number>,<bumber>)

## Example:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     #rgb-color
7     {
8       color: □ rgb(0,0,255);
9     }
10  </style>
11 </head>
12 <body>
13   <p id="rgb-color">This is an Hexadecimal
14     color in CSS</p>
15 </body>
16 </html>
```

This is an Hexadecimal color in CSS

# CSS BACKGROUNDS

Element background can be filled with a color or image, clipped and/or resized, and otherwise be modified.

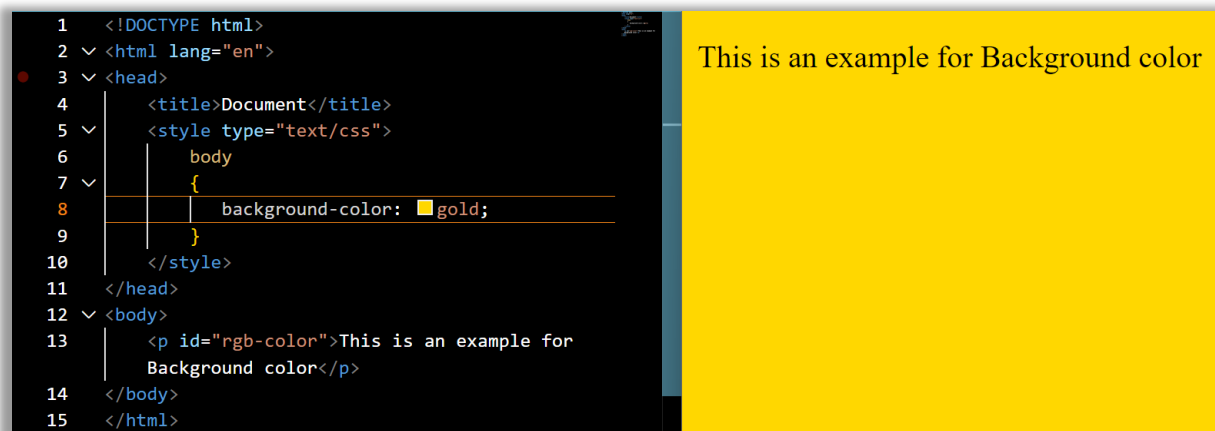
Types of Background Properties:

1. Background-color
2. Background-image
3. Background-repeat.
4. Background-attachment
5. Background-position

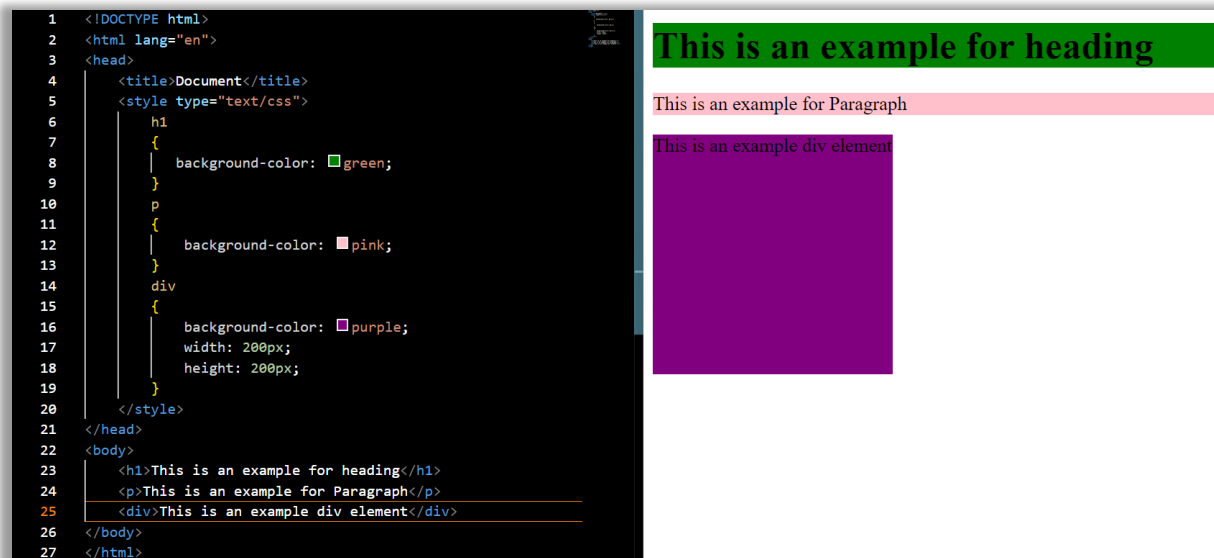
## Background-color:

We can specify a background color for an element using the background-color property.

Example script for changing the background color of a whole page:



Example script for changing the background color of elements:

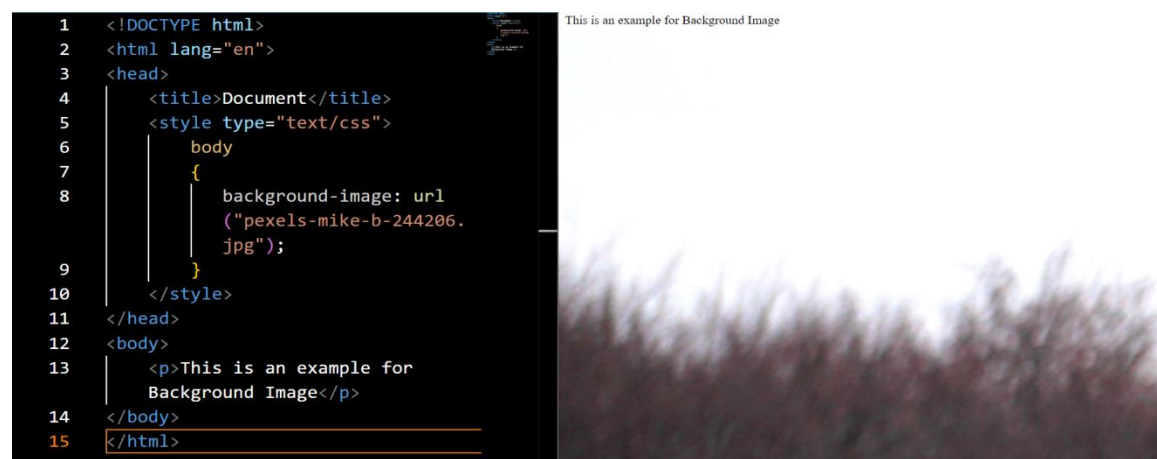


## Background Image

The background-image property sets one or more images as background of an element.

The format of its value should be: url("image.jpg"). Single quote and no quote also work.

### Example:



## Background Image Repeat

CSS automatically resets background images horizontally and vertically.

To only repeat the background image horizontally or vertically we can use background-repeat property.

### Example for Horizontal repeat:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Document</title>
5  |   <style type="text/css">
6  |       body
7  |       {
8  |           background-image: url("pexels-mike-b-244206.jpg");
9  |           background-repeat: repeat-x;
10 |       }
11 |   </style>
12 </head>
13 <body>
14 |   <p>This is an example for Horizontal repeat Image</p>
15 </body>
16 </html>
```

### Example for Vertical repeat:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Document</title>
5  |   <style type="text/css">
6  |       body
7  |       {
8  |           background-image: url("pexels-mike-b-244206.jpg");
9  |           background-repeat: repeat-y;
10 |       }
11 |   </style>
12 </head>
13 <body>
14 |   <p>This is an example for Vertical repeat Image</p>
15 </body>
16 </html>
```



## Background Image no-repeat

The no-repeat value of the background-repeat property stops a background image from repeating.

### Example for no-repeat:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5      <style type="text/css">
6          body
7          {
8              background-image: url("pexels-mike-b-244206.jpg");
9              background-repeat: no-repeat;
10         }
11     </style>
12 </head>
13 <body>
14     <p>This is an example for No repeat Image</p>
15 </body>
16 </html>
```

## Background Image Set Position

We can set the initial position of a background image using the background-position property.

### Example:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5      <style type="text/css">
6          body
7          {
8              background-image: url("pexels-mike-b-244206.jpg");
9              background-repeat: no-repeat;
10             background-position: top;
11         }
12     </style>
13 </head>
14 <body>
15     <p>This is an example for Top Background Image</p>
16 </body>
17 </html>
```

## Background Attachments:

The background-attachment property sets whether a background image's position is fixed within the viewport, or scrolls with its containing blocks.

### Example:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5      <style type="text/css">
6          body
7          {
8              background-image: url("pexels-mike-b-244206.jpg");
9              background-attachment: scroll;
10         }
11         p
12         {
13             font-size: 200px;
14             color: black;
15         }
16     </style>
17 </head>
18 <body>
19     <p>This is an example for Top Background attachments</p>
20 </body>
21 </html>
```

# CSS3 MEDIA QUERIES

CSS Media Queries are useful when you want to modify your web page depending on a device's general type or specific characteristics and parameters.

- Width and height of the viewport.
- Orientation (portrait and landscape)
- Screen resolution.

A media query is composed of an optional media type and any number of media feature expressions.

It can have multiple expressions combined using logical operators. Media queries are case-insensitive.

A media query computes to true when the media type matches the device. When that happens, all corresponding style sheets or rules will be applied.

## **CSS Media Types:**

CSS media types describe the category of a device:

- all: suitable for all the devices.
- print: intended for paged material and documents viewed on a screen in print preview mode.

- screen: intended primarily for screens.
- speech: intended for speech synthesizers.

## **CSS Media Features:**

CSS Media features describes specific characteristics of the user agent, output devices or environment.

## **The @media Rule:**

The @media at-rule is used to target media types and media features.

## **Syntax:**

```
@media not| only mediatype and (expressions)
{
  CSS-Rules;
}
```

Here the expression resolve to either true or false.

If true, it means that the expressions match the media type of the device the web page is being displayed on.

When that is the case then all the corresponding style sheet (e.g. CSS declaration blocks) are applied to the document.

The media type is optional except when using the not or only operators. If they are not used, the media type will be implied.

## Example:


The below script automatically change the background color of the document depending on the media features and screen size of the device.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5   <style type="text/css">
6     body
7     {
8       background-color: tan;
9       color: black;
10    }
11    @media screen and (max-width: 992px)
12    {
13      body
14      {
15        background-color: blue;
16        color: white;
17      }
18    }
19    @media screen and (max-width: 600px)
20    {
21      body
22      {
23        background-color: olive;
24        color: white;
25      }
26    }
27  </style>
28 </head>
29 <body>
30   <h1>Resize the browser window to see the effect!</h1>
31   <p>By default, the background color of the document is "tan". If the screen size is 992px or less, the color will change to "blue". If it is 600px or less, it will change to "olive".</p>
32 </body>
33 </html>
```

## Default size:

**Resize the browser window to see the effect!**


By default, the background color of the document is "tan". If the screen size is 992px or less, the color will change to "blue". If it is 600px or ledd, it will change to "olive".



## @ 992px or less:

**Resize the browser window to see the effect!**

By default, the background color of the document is "tan". If the screen size is 992px or less, the color will change to "blue". If it is 600px or ledd, it will change to "olive".



**@ 600px or less:**

**Resize the browser window to see the effect!**

By default, the background color of the document is "tan". If the screen size is 992px or less, the color will change to "blue". If it is 600px or less, it will change to "olive".