UNIT-1

1.What is a web browser?

- It is an application that we use when we browse the WWW.
- It renders text-based HTML documents into visual pages, which are what we see inside a browser.
- It speaks HTTP protocol and communicates with web servers.
- It understands URL and knows how to translates URL into web resources, e.g., HTML text files, images, videos, etc.
- It is a virtual machine that runs the JavaScript programs embedded inside HTML documents.
- It understands CSS rules and applies the rules to layout the pages.
- It interacts with a user in front of a browser and translates user inputs into browser events, e.g., clicking a link, clicking a button, submitting text inside a text box.
- It makes the WWW come alive!

Browser functionality

- caching: keep local copies of documents that have been downloaded and viewed before;
- authentication: validate the credential of the user when visiting a secure web site;
- state maintenance: keep "cookies" within a session across many request/response transactions;
- on-demand downloading: as an HTML document is being rendered, missing items (e.g., CSS style sheets, JavaScript code, images, audios, videos, etc.) will be loaded on demand;
- header processing: the header of an HTML document may contain "redirection";
- content type processing: process the embedded content (i.e., image formats, audio formats, video formats) accordingly;
- handling errors.

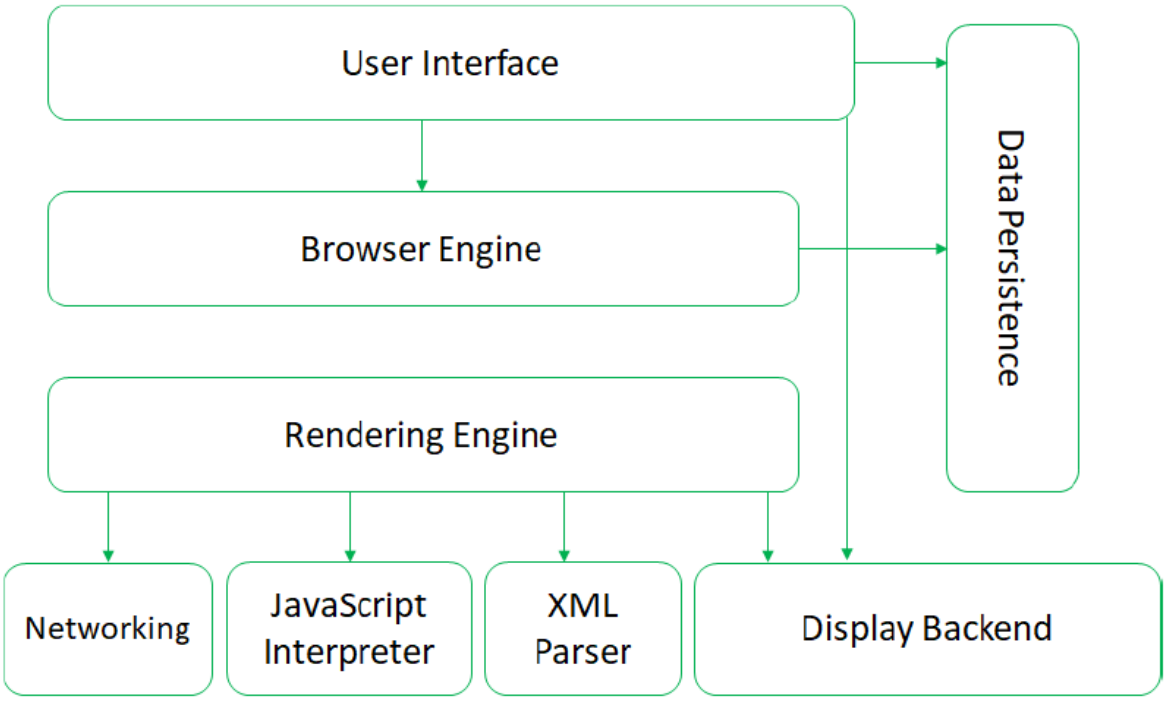
2.Explain Browser Functionality and Architecture.

Browser Architecture

The Web Browser architecture is shown in below Figure. It has eight major subsystems and their dependencies:

1. The User Interface subsystem is the layer between the user and the Browser Engine. It provides features such as toolbars, visual page-load progress, smart download handling, preferences, and printing. It may be integrated with the desktop environment to provide browser session management or communication with other desktop applications.

2. The Browser Engine subsystem is an embedded component that provides a high-level interface to the Rendering Engine(Which is the next layer). It loads a given URL and supports primitive browsing actions such as forward, back, and reload. It provides hooks for viewing various aspects of the browsing session such as current page load progress and JavaScript alerts. It also allows the querying and manipulation of Rendering Engine settings.



3. The Rendering Engine subsystem produces a visual representation for a given URL. It is capable of displaying HTML and Extensible Markup Language (XML) documents, optionally styled with CSS, as well as embedded content such as images.

4. The Networking subsystem implements file transfer protocols such as HTTP and FTP

5. The JavaScript Interpreter evaluates JavaScript code, which may be embedded in web pages.

6. The XML Parser subsystem parses XML documents into a Document Object Model (DOM) tree. This is one of the most reusable subsystems in the architecture. In fact, almost all browser implementations leverage an existing XML Parser rather than rewriting their own from scratch.

7.The Display Back-end subsystem provides drawing and windowing primitives, a set of user interface widgets, and a set of fonts. It may be tied closely with the operating system.

8. The Data Persistence subsystem stores various data associated with the browsing session on disk. This may be high-level data such as bookmarks or toolbar settings, or it may be low-level data such as cookies, security certificates, or cache.

3. HTML5 and its features.

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

Browser Support

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- **WebSocket** − A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.
- **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

4. Write about HTML- basic Tags.

The **<!DOCTYPE html>** statement must always be the first to appear on an HTML page and tells the browser which version of the language is being used. In this case, we are working with HTML5.

The <html> and </html> tags tell the web browser where the HTML code starts and ends.

The HTML <head> Element

The HTML <head> element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed.

The <head> element is placed between the <html> tag and the <body> tag:

HTML Headings

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
```

<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>

```
<!DOCTYPE html>
<html>
<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>
<body>
<p>The HTML head element contains meta data.</p>
<p>Meta data is data about the HTML document.</p>
</body>
</html>
```

HTML Paragraphs
The HTML element defines a paragraph:

Example
This is a paragraph.
This is another paragraph.

HTML Line Breaks
The HTML <br> element defines a line break.
Use <br> if you want a line break (a new line) without starting a new paragraph:

The HTML <pre> Element
The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

HTML Formatting Elements
In the previous chapter, you learned about the HTML style attribute.

HTML also defines special elements for defining text with a special meaning.

HTML uses elements like <b> and <i> for formatting output, like bold or italic text.

Formatting elements were designed to display special types of text:

<b> - Bold text
 - Important text
<i> - Italic text
 - Emphasized text
<mark> - Marked text
<small> - Small text
<del> - Deleted text

<ins> - Inserted text
<sub> - Subscript text
<sup> - Superscript text

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. It can be an image or any other HTML element.

HTML Links - Syntax

Hyperlinks are defined with the HTML `<a>` tag: `<a href="url">link text</a>`

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

HTML Images

Images can improve the design and the appearance of a web page.

HTML Images Syntax

In HTML, images are defined with the `<img>` tag.

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.

The `src` attribute specifies the URL (web address) of the image:

```
<img src="url">
<img src="img_chania.jpg" alt="Flowers in Chania">
```

5.Discuss about Lists in HTML.

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

A Description List

Coffee
- black hot drink

Milk
- white cold drink

6.Discuss about Table Tags.

Defining an HTML Table

An HTML table is defined with the `<table>` tag.

Each table row is defined with the `<tr>` tag. A table header is defined with the `<th>` tag. By default, table headings are bold and centered. A table data/cell is defined with the `<td>` tag.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS `border` property:

Example

```
table, th, td {
  border: 1px solid black;
}
```

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the `colspan` attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the `rowspan` attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

7.Write about HTML Frames.

The <frameset>.

- The <frameset> tag defines a frameset.
- The <frameset> element holds one or more <frame> elements. Each <frame> element can hold a separate document.
- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

Attribute	Value	Description
cols	Pixels % *	It specifies the number and size of column spaces in the frameset. (Not Supported in HTML5)
rows	Pixels % *	It specifies the number and size of the rows spaces in the frameset. (Not Supported in HTML5)

Tag-specific attribute

Attribute	Value	Description
frameborder	0 1	It specifies whether to display a border around the frame or not, and its default value is 1
longdsec	URL	It specifies a page which contains the long description of the content of the frame.
marginheight	pixels	It specifies the top and bottom margins of the frame.
marginwidth	pixels	It defines the height of the margin between frames.
name	text	It is used to assign the name to the frame.
noresize	noresize	It is used to prevent resizing of the frame by the user.
scrolling	yes no auto	It specifies the existence

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.    <title>Frame tag</title>
5. </head>
6.  <frameset cols="50%,50%">
```

```
7.     <frame  src="https://www.javatpoint.com/html-table">
8.     <frame  src="https://www.javatpoint.com/css-table">
9.    </frameset>
10. </html>
```

HTML iframes

HTML Iframe is used to display a nested webpage (a webpage within a webpage). The HTML <iframe> tag defines an inline frame, hence it is also called as an Inline frame.

An HTML iframe embeds another document within the current HTML document in the rectangular region.
The webpage content and iframe contents can interact with each other using JavaScript.

Iframe Syntax

An HTML iframe is defined with the <iframe> tag:

<iframe src="URL"></iframe>

Here, "src" attribute specifies the web address (URL) of the inline frame page.

Set Width and Height of iframe

You can set the width and height of iframe by using "width" and "height" attributes. By default, the attributes values are specified in pixels but you can also set them in percent. i.e. 50%, 60% etc.

```
<html>
<body>
<h2>HTML Iframes example</h2>
<p>Use the height and width attributes to specify the size of the iframe:</p>
<iframe src="https://www.javatpoint.com/" height="300" width="400"></iframe>
</body>
</html>
```

8.Write about HTML-5 tags.

Ans: add basic tags here. Give in question-4

HTML Audio Tag

HTML audio tag is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3

2. wav
3. ogg

HTML Audio - How It Works

The `controls` attribute adds audio controls, like play, pause, and volume.

The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

HTML Audio Tag Example

Let's see the code to play mp3 file using HTML audio tag.

```
<audio controls>
  <source src="koyal.mp3" type="audio/mpeg">
Your browser does not support the html audio tag.
</audio>
```

The HTML <video> Element

To show a video in HTML, use the <video> element:

How it Works

The `controls` attribute adds video controls, like play, pause, and volume.

If height and width are not set, the page might flicker while the video loads.

The `<source>` element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

Example

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
```

</video>

The HTML `<canvas>`

The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>
</body>
</html>
```

HTML iframe Tag

An iframe is used to display a web page within a web page.

- An HTML iframe is defined with the <iframe> tag:

<iframe src="*URL*"></iframe>

The src attribute specifies the URL (web address) of the inline frame page.

- Iframe - Set Height and Width

Use the height and width attributes to specify the size of the iframe.
The height and width are specified in pixels by default:
Example
<iframe src="demo_iframe.htm" height="200" width="300"></iframe>

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

HTML <map> tag

HTML <map> tag is used with <area> tag to define a client-side image map.

An image map is consist of an image with clickable areas, where you can click on the image, and it will open to new or the provided destination.

The <map> tag can consist of more than one <area> elements which define the coordinates and type of the area.

```
1. <img src="image1.png" usemap="#web">
2. <map name="web">
3. <area shape="rect" coords="66,117,131,168" href="https://www.javatpoint.com/html-tutorial">
```

9.Write about HTML 5 Forms and elements.

The <form> Element

The HTML `<form>` element defines a form that is used to collect user input:

```
<form>
.
form elements
.
</form>
```

An HTML form contains **form elements**.

Form elements are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

The `<input>` element is the most important form element.

The `<input>` element is displayed in several ways, depending on the **type** attribute.

Here are some examples:

These input elements use the **type** attribute to specify the data type.HTML4 provides following types – also in HTML 5.

Sr.No.	Type & Description
1	**Text** A free-form text field, nominally free of line breaks.
2	**Password** A free-form text field for sensitive information, nominally free of line breaks.
3	**Checkbox** A set of zero or more values from a predefined list.
4	**Radio** An enumerated value.
5	**Submit** A free form of button initiates form submission.
6	**File** An arbitrary file with a MIME type and optionally a file name.
7	**Image** A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form submission.
8	**Hidden** An arbitrary string that is not normally displayed to the user.
9	**Select** An enumerated value, much like the radio type.
10	**Textarea** A free-form text field, nominally with no line break restrictions.
11	**Button** A free form of button which can initiates any event related to button.

Example:-

```
<form action = "http://example.com/cgiscript.pl" method = "post">

  <p>

    <label for = "firstname">first name: </label>

    <input type = "text" id = "firstname"><br />

      <label for = "lastname">last name: </label>

    <input type = "text" id = "lastname"><br />

       <label for = "email">email: </label>

    <input type = "text" id = "email"><br>

       <input type = "radio" name = "sex" value = "male"> Male<br>

    <input type = "radio" name = "sex" value = "female"> Female<br>

    <input type = "submit" value = "send"> <input type = "reset">
```

</p> </form>

Newly Defined HTML 5 Elements in Form

Sr.No.	Type & Description
1	datetime A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.
2	datetime-local A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.
3	date A date (year, month, day) encoded according to ISO 8601.
4	month A date consisting of a year and a month encoded according to ISO 8601.
5	week A date consisting of a year and a week number encoded according to ISO 8601.
6	time A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.
7	number It accepts only numerical value. The step attribute specifies the precision, defaulting to 1.
8	range The range type is used for input fields that should contain a value from a range of numbers.
9	email It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.
10	url It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in http://www.example.com format or in http://example.com format.

```
<!DOCTYPE HTML>
<html>

   <body>

      <form action = "/cgi-bin/html5.cgi" method = "get">
         Date and Time : <input type = "datetime" name = "newinput"
/>
Enter email : <input type = "email" name = "newinput" />
         <input type = "submit" value = "submit" />
      </form>
```

```
   </body>
</html>
```

10 What is CSS and how are they linked with HTML.

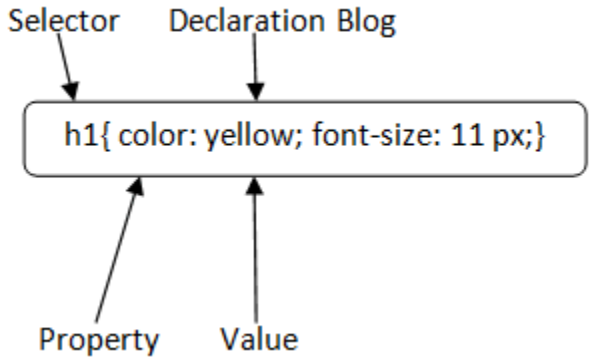
CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax

A CSS rule set contains a selector and a declaration block.



Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is a Selector{Property1: value1; Property2: value2; .....;} ssigned to color property.

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

1) Inline CSS

In For example:

** Hello CSS **

2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:line CSS is used to apply CSS on a single line or element.

1. **<style>**
2. p{color:blue}
3. **</style>**

3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

For example: p{color:blue}

You need to link this style.css file to your html pages like this:

<link rel="stylesheet" type="text/css" href="style.css"**>**

1. All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:
2. Inline style (inside an HTML element)
3. External and internal style sheets (in the head section)
4. Browser default

5. So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

11.Write about CSS selectors.

CSS selectors are used to "find" (or select) the HTML elements you want to style.

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

```
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

```
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1>Hello world!</h1>
<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

12.List out CSS properties .

CSS Background Color

You can set the background color for HTML elements:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>

body {
  background-color: lightblue;
}
```

CSS Text Color

You can set the color of text:

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

CSS Border Color

You can set the color of border

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

CSS Border Style

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border
- `groove` - Defines a 3D grooved border. The effect depends on the border-color value
- `ridge` - Defines a 3D ridged border. The effect depends on the border-color value

CSS background-image

The `background-image` property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

```
body {
  background-image: url("paper.gif");
}
```

CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS Padding

The CSS `padding` properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- `padding-top`
- `padding-right`
- `padding-bottom`

- `padding-left`

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Example

```
a {
 color: hotpink;
}
```

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

- `a:link` - a normal, unvisited link

- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

```
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
```

CSS Layout - The Display Property

The `display` property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is `block` or `inline`.

Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The element is a block-level element.

Examples of block-level elements:

-

- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

Display: none;
display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The <script> element uses display: none; as default.

```
span {
  display: block;
}
```

13. List our some CSS properties for Text Formatting.

Text Color

The color property is used to set the color of the text. The color is specified by:
a color name - like "red"
a HEX value - like "#ff0000"
an RGB value - like "rgb(255,0,0)"
The default text color for a page is defined in the body selector.

Example

```
body {
 color: blue;
}
h1 {
 color: green;
}
```

Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

```
h1 {
  text-align: center;
```

```
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
```

Text Direction

The direction and unicode-bidi properties can be used to change the text direction of an element:

```
Example

p {
 direction: rtl;
 unicode-bidi: bidi-override;
}
```

Text Decoration

The text-decoration property is used to set or remove decorations from text.

```
h1 {
 text-decoration: overline;
}
h2 {
 text-decoration: line-through;
}
h3 {
 text-decoration: underline;
}
```

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text:

```
Example

p {
 text-indent: 50px;
}
```

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text.
The following example demonstrates how to increase or decrease the space between characters:
Example

```
h1 {
  letter-spacing: 3px;
}

h2 {
  letter-spacing: -3px;
}
```

CSS Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {
  font-style: normal;
}
p.italic {
  font-style: italic;
}
p.oblique {
  font-style: oblique;
}
```

Font Weight

The font-weight property specifies the weight of a font:

Example

```
p.normal {
  font-weight: normal;
}
p.thick {
  font-weight: bold;
}
```

Font Size

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Example

```
h1 {
  font-size: 40px;
}

h2 {
  font-size: 30px;
}

p {
  font-size: 14px;
}
```

14.List our CSS TABLE formatting properties.

Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for <table>, <th>, and <td> elements:

```
table, th, td {
  border: 1px solid black;
}
```

Collapse Table Borders

The border-collapse property sets whether the table borders should be collapsed into a single border:

```
table {
  border-collapse: collapse;
}

table, th, td {
  border: 1px solid black;
}
```

Table Width and Height

Width and height of a table are defined by the width and height properties.

```
table {
  width: 100%;
}

th {
  height: 50px;
}
```

Horizontal Alignment

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

```
th {
  text-align: left;
}
```

Vertical Alignment

The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).

```
td {
  height: 50px;
  vertical-align: bottom;
}

th {
  background-color: #4CAF50;
  color: white;
}
```