**What is HTML? And Explain in detail.**

HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
* Browsers do not display the HTML tags, but use them to render the content of the page

**Example program**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**New HTML5 Elements**

The most interesting new HTML5 elements are:

* New semantic elements like <header>, <footer>, <article>, and <section>.
* New attributes of form elements like number, date, time, calendar, and range.
* New graphic elements: <svg> and <canvas>.
* New multimedia elements: <audio> and <video>.

**New HTML5 API's (Application Programming Interfaces)**

The most interesting new API's in HTML5 are:

* HTML Geolocation
* HTML Drag and Drop
* HTML Local Storage
* HTML Application Cache
* HTML Web Workers
* HTML SSE

|  |  |
| --- | --- |
| **Removed Element** | **Use Instead** |
| <acronym> | <abbr> |
| <applet> | <object> |
| <basefont> | CSS |
| <big> | CSS |
| <center> | CSS |
| <dir> | <ul> |
| <font> | CSS |
| <frame> |  |
| <frameset> |  |
| <noframes> |  |
| <strike> | CSS, <s>, or <del> |
| <tt> | CSS |

## Add New Elements to HTML

You can also add new elements to an HTML page with a browser trick. This example adds a new element called <myHero> to an HTML page, and defines a style for it:

<!DOCTYPE html>  
<html>  
<head>  
<script>document.createElement("myHero")</script>  
<style>  
myHero {  
    display: block;  
    background-color: #dddddd;  
    padding: 50px;  
    font-size: 30px;  
}   
</style>   
</head>  
<body>  
<h1>A Heading</h1>  
<myHero>My Hero Element</myHero>  
</body>  
</html>

## HTML5Shiv Example

If you do not want to download and store the HTML5Shiv on your site, you could reference the version found on the CDN site. The HTML5Shiv script must be placed in the <head> element, after any style sheets:

### Example

<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<!--[if lt IE 9]>  
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
<![endif]-->  
</head>  
<body>  
<section>  
<h1>Famous Cities</h1>  
<article>  
<h2>London</h2>  
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  
</article>  
<article>  
<h2>Paris</h2>  
<p>Paris is the capital and most populous city of France.</p>  
</article>  
<article>  
<h2>Tokyo</h2>  
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>  
</article>  
</section>  
</body>  
</html>

**New Semantic/Structural Elements**

HTML5 offers new elements for better document structure:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<article>](https://www.w3schools.com/tags/tag_article.asp) | Defines an article in a document |
| [<aside>](https://www.w3schools.com/tags/tag_aside.asp) | Defines content aside from the page content |
| [<bdi>](https://www.w3schools.com/tags/tag_bdi.asp) | Isolates a part of text that might be formatted in a different direction from other text outside it |
| [<details>](https://www.w3schools.com/tags/tag_details.asp) | Defines additional details that the user can view or hide |
| [<dialog>](https://www.w3schools.com/tags/tag_dialog.asp) | Defines a dialog box or window |
| [<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp) | Defines a caption for a <figure> element |
| [<figure>](https://www.w3schools.com/tags/tag_figure.asp) | Defines self-contained content |
| [<footer>](https://www.w3schools.com/tags/tag_footer.asp) | Defines a footer for a document or section |
| [<header>](https://www.w3schools.com/tags/tag_header.asp) | Defines a header for a document or section |
| [<main>](https://www.w3schools.com/tags/tag_main.asp) | Defines the main content of a document |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| [<meter>](https://www.w3schools.com/tags/tag_meter.asp) | Defines a scalar measurement within a known range (a gauge) |
| [<nav>](https://www.w3schools.com/tags/tag_nav.asp) | Defines navigation links |
| [<progress>](https://www.w3schools.com/tags/tag_progress.asp) | Represents the progress of a task |
| [<rp>](https://www.w3schools.com/tags/tag_rp.asp) | Defines what to show in browsers that do not support ruby annotations |
| [<rt>](https://www.w3schools.com/tags/tag_rt.asp) | Defines an explanation/pronunciation of characters (for East Asian typography) |
| [<ruby>](https://www.w3schools.com/tags/tag_ruby.asp) | Defines a ruby annotation (for East Asian typography) |
| [<section>](https://www.w3schools.com/tags/tag_section.asp) | Defines a section in a document |
| [<summary>](https://www.w3schools.com/tags/tag_summary.asp) | Defines a visible heading for a <details> element |
| [<time>](https://www.w3schools.com/tags/tag_time.asp) | Defines a date/time |
| [<wbr>](https://www.w3schools.com/tags/tag_wbr.asp) | Defines a possible line-break |

**Semantic Elements in HTML5**

Below is an alphabetical list of the new semantic elements in HTML5. The links go to our complete [HTML5 Reference](https://www.w3schools.com/tags/default.asp).

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<article>](https://www.w3schools.com/tags/tag_article.asp) | Defines an article |
| [<aside>](https://www.w3schools.com/tags/tag_aside.asp) | Defines content aside from the page content |
| [<details>](https://www.w3schools.com/tags/tag_details.asp) | Defines additional details that the user can view or hide |
| [<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp) | Defines a caption for a <figure> element |
| [<figure>](https://www.w3schools.com/tags/tag_figure.asp) | Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| [<footer>](https://www.w3schools.com/tags/tag_footer.asp) | Defines a footer for a document or section |
| [<header>](https://www.w3schools.com/tags/tag_header.asp) | Specifies a header for a document or section |
| [<main>](https://www.w3schools.com/tags/tag_main.asp) | Specifies the main content of a document |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| [<nav>](https://www.w3schools.com/tags/tag_nav.asp) | Defines navigation links |
| [<section>](https://www.w3schools.com/tags/tag_section.asp) | Defines a section in a document |
| [<summary>](https://www.w3schools.com/tags/tag_summary.asp) | Defines a visible heading for a <details> element |
| [<time>](https://www.w3schools.com/tags/tag_time.asp) | Defines a date/time |

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>HTML5</title>

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

</script>

<![endif]-->

<style>

body {

font-family: Verdana,sans-serif;

font-size: 0.9em;

}

header, footer {

padding: 10px;

color: white;

background-color: black;

}

section {

margin: 5px;

padding: 10px;

background-color: lightgrey;

}

article {

margin: 5px;

padding: 10px;

background-color: white;

}

nav ul {

padding: 0;

}

nav ul li {

display: inline;

margin: 5px;

}

</style>

</head>

<body>

<header>

<h1>Monday Times</h1>

</header>

<nav>

<ul>

<li>News</li>

<li>Sports</li>

<li>Weather</li>

</ul>

</nav>

<section>

<h2>News Section</h2>

<article>

<h2>News Article</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>

</article>

<article>

<h2>News Article</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>

</article>

</section>

<footer>

<p>&copy; 2014 Monday Times. All rights reserved.</p>

</footer>

</body>

</html>

**Use Lower Case Element Names**

HTML5 allows mixing uppercase and lowercase letters in element names.

We recommend using lowercase element names because:

* Mixing uppercase and lowercase names is bad
* Developers normally use lowercase names (as in XHTML)
* Lowercase look cleaner
* Lowercase are easier to write

## Omitting <html> and <body>?

In HTML5, the <html> tag and the <body> tag can be omitted.

The following code will validate as HTML5:

**Example Program:**

<!DOCTYPE html>

<head>

<title>Page Title</title>

</head>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

## What is HTML Canvas?

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. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

**Canvas Examples**

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

**Note:** Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute. Here is an example of a basic, empty canvas:

### Draw a Line

<!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>

### Draw a Circle

<!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.beginPath();

ctx.arc(95,50,40,0,2\*Math.PI);

ctx.stroke();

</script>

</body>

</html>

### Stroke Text

<!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.font = "30px Arial";

ctx.strokeText("Hello World",10,50);

</script>

</body>

</html>

### Draw Circular Gradient

<!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");

// Create gradient

var grd = ctx.createLinearGradient(0,0,200,0);

grd.addColorStop(0,"red");

grd.addColorStop(1,"white");

// Fill with gradient

ctx.fillStyle = grd;

ctx.fillRect(10,10,150,80);

</script>

</body>

</html>

**What is SVG?**

* SVG stands for Scalable Vector Graphics
* SVG is used to define graphics for the Web
* SVG is a W3C recommendation

The HTML <svg> Element, The HTML <svg> element is a container for SVG graphics, SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

**Circle Program**

<!DOCTYPE html>

<html>

<body>

<svg width="100" height="100">

<circle cx="50" cy="50" r="40"

stroke="green" stroke-width="4" fill="yellow" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

## SVG Rectangle

<!DOCTYPE html>

<html>

<body>

<svg width="400" height="100">

<rect width="400" height="100"

style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

**Star Program**

<!DOCTYPE html>

<html>

<body>

<svg width="300" height="200">

<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

## Differences Between SVG and Canvas

SVG is a language for describing 2D graphics in XML., Canvas draws 2D graphics, on the fly (with a JavaScript)., SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

**Video and audio tags**

<!DOCTYPE html>

<html>

<body>

<div style="text-align:center">

<button onclick="playPause()">Play/Pause</button>

<button onclick="makeBig()">Big</button>

<button onclick="makeSmall()">Small</button>

<button onclick="makeNormal()">Normal</button>

<br><br>

<video id="video1" width="420">

<source src="mov\_bbb.mp4" type="video/mp4">

<source src="mov\_bbb.ogg" type="video/ogg">

Your browser does not support HTML5 video.

</video>

</div>

<script>

var myVideo = document.getElementById("video1");

function playPause() {

if (myVideo.paused)

myVideo.play();

else

myVideo.pause();

}

function makeBig() {

myVideo.width = 560;

}

function makeSmall() {

myVideo.width = 320;

}

function makeNormal() {

myVideo.width = 420;

}

</script>

<p>Video courtesy of <a href="https://www.bigbuckbunny.org/" target="\_blank">Big Buck Bunny</a>.</p>

</body>

</html>

**What is CSS? And Explain in detail.**

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External style sheets are stored in CSS files

**Why Use CSS?**

* 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 Solved a Big Problem**

* HTML was NEVER intended to contain tags for formatting a web page!
* HTML was created to **describe the content** of a web page, like:
* <h1>This is a heading</h1>
* <p>This is a paragraph.</p>
* When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
* To solve this problem, the World Wide Web Consortium (W3C) created CSS.
* CSS removed the style formatting from the HTML page!

CSS Saves a Lot of Work!

* The style definitions are normally saved in external .css files.
* With an external stylesheet file, you can change the look of an entire website by changing just one file!

**CSS Syntax**

A CSS rule-set consists of a selector and a declaration block:



* The selector points to the HTML element you want to style.
* The declaration block contains one or more declarations separated by semicolons.
* Each declaration includes a CSS property name and a value, separated by a colon.
* A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

**CSS Selectors**

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

**The element Selector**

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

**The id Selector**

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

The id of an element should be 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.

The style rule below will be applied to the HTML element with id="para1":

**The class Selector**

* The class selector selects elements with a specific class attribute.
* To select elements with a specific class, write a period (.) character, followed by the name of the class.
* In the example below, all HTML elements with class="center" will be red and center-aligned:

**Grouping Selectors**

* If you have elements with the same style definitions, like this:
* It will be better to group the selectors, to minimize the code.
* To group selectors, separate each selector with a comma.
* In the example below we have grouped the selectors from the code above:

**CSS Comments**

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. A CSS comment starts with /\* and ends with \*/. Comments can also span multiple lines:

**Three Ways to Insert CSS**

There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

**External Style Sheet**

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

**Internal Style Sheet**

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

**Inline Styles**

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property. The example below shows how to change the color and the left margin of a <h1> element:

**Cascading Order**

What style will be used when there is more than one style specified for an HTML element? Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

**Example Program:**

<!DOCTYPE html>

<html>

<head>

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

<style>

body {background-color: linen;}

</style>

</head>

<body style="background-color: lightcyan">

<h1>Multiple Styles Will Cascade into One</h1>

<p>In this example, the background color is set inline, in an internal stylesheet, and in an external stylesheet.</p>

<p>Try experimenting by removing styles to see how the cascading stylesheets work. (try removing the inline first, then the internal, then the external)</p>

</body>

</html>