

```
<html lang="en">
```

```
<head>
```

```
<meta
```

```
<meta
```

```
<title>
```

```
</head>
```

```
<body>
```

```
<h1>Page content</h1>
```

HTML



History of HTML



- **HTML** was created by Sir Tim Berners-Lee in late 1991 but was not released officially, which was published in 1995 as **HTML 2.0**.
- HTML is developed and maintained by world wide web consortium known as W3C

Requirements



- Notepad
- Web Browser

What Stands For HTML ?



- HTML stands for Hyper Text Markup Language
- **Hypertext:** Hypertext is text that provides a link to other text, which can be located on another web page, on the same page, or within a document. Ted Nelson was the first to use this term.
- **Markup:** A markup language provides a way to describe the structure of text and graphics on web pages.

`<html>`

`<html>` tag indicates that this web page is written in HTML.

`<head>`

`<title>` Title of the web page `</title>`

`<title>` tag contains the web page title.

`</head>`

`<head>` tag contains information about the web.

`<body>`

Content of the page.

`<body>` tag contains the content of the web page.

`</body>`

`</html>`

`</html>` marks the end of the web page.

Scripts



- A *script* is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be Javascript or VBScript.
- * **Attribute:** an **attribute** is a characteristic of a page element, such as font size or color. **Attributes** are used to amplify a **tag**. When a Web browser interprets an **HTML tag**, it will also look for its **attributes** so that it can display the Web page's elements properly

- `<html>`
- `<head>`
- `<title>Internal Script</title>`
- `</head>`
- `<body>`
- `<script type="text/javascript">`
`document.write("Hello Javascript!") </script>`
`</body>`
- `</html>`

- <head>
- <title>Event Handler Example t</title>
- <script type="text/javascript">
- function myAlert()
- {
- alert("I am an event handler....");
- return;
- }
- </script>
- </head>
- <body>
- <p onmouseover="myAlert();">
- Bring your mouse here to see an alert
- </p>
- </body>



- `<!DOCTYPE html>`
- `<html>`
-
- `<head>`
- `<title>HTML CSS</title>`
- `</head>`
-
- `<body>`
- `<p>Hello,`
`World!</p>`
- `</body>`
-
- `</html>`

Hide Scripts from Older Browsers



- though most (if not all) browsers these days support scripts, some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this from happening, you can simply place HTML comments around the script. Older browsers will ignore the script, while newer browsers will run it.



- `<script type="text/javascript">`
- `<!-- document.write("Hello Javascript!");`
- `-->`
- `</script>`



- Basic Idea About Attributes

Noscript:



- You can also provide alternative info for users whose browsers don't support scripts and for users who have disabled scripts. You do this using the `<noscript>` tag.



- `<script type="text/javascript">`
- `<!-- document.write("Hello Javascript!");`
- `--> </script>`
- `<noscript>Your browser does not support Javascript!</noscript>`



<Meta> Tag

- The <meta> tag in HTML provides information about HTML Document or in simple words, it provides important information about a document. These tags are basically used to add name/value pairs to describe properties of HTML document, such as expiry date, author name, list of keywords, document author, etc. This tag is an empty element because it only has an opening tag and no closing tag but it carries information within its attributes. A web document can include one or more meta tag depending on information, but in general, it doesn't affect the physical appearance of the document.

- **Specifying Keywords: Keywords** are a specific type of **meta tag** that appear in the HTML code of a Web page and help tell search engines what the topic of the page is..
- `<html>`
- `<head>`
- `<title>meta tag examples</title>`
- `<meta name = "keywords" content = "Meta Tags, Metadata"/>`
- `</head>`
- `<body>`
- `<p>Hello world!</p>`
- `</body>`
- `</html>`



- **Document Description:** This meta tag is used in HTML document to provide short description about the document

- <!DOCTYPE html>
- <html>
- <head>
- <title>Meta Tags example</title>
- <meta name = "keywords" content = "Meta Tags, Metadata" />
- <meta name = "description"
- content = "this is a computer science portal." />
- </head>
- <body>
- <p>Hello World!!!</p>
- </body>
- </html>



- **Document Revision Date:** This meta tag is used to give the information about last updated document. This information is used by various web browser when refresh web page.

- <!DOCTYPE html>
- <html>
- <head>
- <title>Meta tags example</title>
- <meta name = "keywords"
- content = "Meta Tags, Metadata" />
- <meta name = "description"
- content = "Learn about Meta Tags." />
- <meta name = "revised detail"
- content = "last updated time" />
- </head>
- <body>
- <p>Hello World!</p>
- </body>
- </html>



- **Setting Author Name:** The name of the author can be set in web page through HTML document using meta tag.

- <!DOCTYPE html>
- <html>
- <head>
- <title>Meta tags example</title>
- <meta name = "keywords used "
- content = "Meta Tags, Metadata" />
- <meta name = "description about"
- content = "Meta tags learning." />
- <meta name = "author" content = "sushant Srivastava" />
- </head>
- <body>
- <p>web development !</p>
- </body>
- </html>

support meta tags are:



- Google chrome
- Internet Explorer
- Mozilla
- Opera
- Safari

What is Viewport?



- A Browser's viewport is the area of web page in which the content is visible to the user. The viewport does not have the same size, it varies with the variation in screen size of the devices on which the website is visible. For a laptop, the viewport has a larger size as compared to a smartphone or tablet.



- **Note:** When a page is not made responsive for smaller viewports it looks bad or even breaks on smaller screen. To fix this problem introduce a responsive tag to control the viewport. This tag was firstly introduced by Apple Inc. for Safari iOS.

A Responsive tags has the following attributed:



- **width:** Width of the virtual viewport of the device.
- **initial-scale:** Zoom level when the page is first visited.

- <html>
- <head>
- <title>hello world</title>
- <meta name="viewport"
- content="width=device-width, initial-scale=1.0">
-
- </head>
- <body>
-
- <p>Prepare for the Recruitment drive of product based companies like
- Microsoft, Amazon, Adobe etc with a free online placement preparation
- course. The course focuses on various MCQ's & Coding question likely
- to be asked in the interviews & make your upcoming placement season
- efficient and successful. </p>
- <p>An extensive Online Test Series for GATE 2019 to boost the
- preparation for GATE 2019 aspirants. Test series is designed
- considering the pattern of previous years GATE papers and ensures
- to resemble with the standard of GATE. This Test Series will help
- the aspirants track and improve the preparation through questions
- of various difficulty levels. There will be two Test Series
- covering the whole syllabus of GATE, including Mathematics and
- Aptitude. Test Series I will cover the basic and medium difficulty,
- whereas in Test Series II difficulty will be slightly higher. </p>
- </body>
- </html>

Translate Attribute



- The translate attribute in HTML is used to specify whether the content of an element is translated or not. This attribute is new in HTML5.
- **Syntax:**
- `<element translate = "yes|no">`

listed below:



- **yes:** This attribute is used to specify that the content of element can be translated.
- **no:** This attribute is used to specify that the content of element can not be translated.



- `<!DOCTYPE html>`
- `<html lang = "fr">`
- `<head>`
- `<title>translate attribute</title>`
- `<style>`
- `</head>`
- `<body>`
- `<h1>Hello</h1>`
- `<h2>translate attribute</h2>`
- `<p translate>`
- `salut comment allez-vous!</p>`
- `<p>This can be translated to any language.</p>`
- `</body>`
- `</html>`

Why it is needed?



- Adding the translate attribute to your page can help readers better understand your content when they run it through automatic translation systems, and can save a significant amount of cost and hassle for translation vendors with large throughput in many languages.
- You come across a need for this quite frequently. There is an example in the HTML5 spec about the Bee Game. Here is a similar, but real example where the documentation being translated referred to a machine with text in English on the hardware panel that wasn't translated.

DOCTYPE



- The HTML document type declaration, also known as DOCTYPE, is the first line of code required in every HTML or XHTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in.



- Doctype syntax for HTML5 and beyond:
- `<!DOCTYPE html>`
- Doctype syntax for strict HTML 4.01:
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

Head



- The **<head>** element is a container for metadata (data about data) and is placed between the **<html>** tag and the **<body>** tag.

- `<!DOCTYPE HTML>`



- `<html>`

- `<head>`

- `<title>hello world</title>`

- `<meta name="viewport"`

- `content= "width=device-width, initial-scale=1.0">`

- `</head>`

- `<body>`

- `helllo こんにちは`

- `</body>`

- `</html>`



- `<!DOCTYPE HTML>`
- `<html>`
- `<head>`
- `<title>hello world</title>`
- `<meta charset=" ISO-8859-1">`
- `<meta name="viewport"`
- `content= "width=device-width, initial-scale=1.0">`
- `</head>`
- `<body>`
- `Helo こんにちは`
- `</body>`
- `</html>`

Specifying your document's character encoding



- The document's character encoding — the character set that the document is permitted to use. utf-8 is a universal character set that includes pretty much any character from any human language. This means that your web page will be able to handle displaying any language; it's therefore a good idea to set this on every web page you create! For example, your page could handle English and Japanese just fine:

<body> tag



- **Description**
- The HTML **<body> tag** defines the main content of the HTML document or the section of the HTML document that will be directly visible on your web page. This tag is also commonly referred to as the **<body> element**.
- **Syntax** `<body> </body>`

BASIC tags



- **<h1>** Put your header here **</h1>** A size 1 header is larger than a size 2. You can begin with any size you choose, designated by h1, h2, h3, for example.
- **<p>** [This tag signifies a paragraph. No final paragraph tag is needed. Final tags, when used, have a "/" in front of the html letters.]
- **
** [This tag at the end of a line signifies a line break.]



- **<center>**Text between these two symbols will be centered on the page.**</center>** For example, if you place these tags before and after the **<H2>**,**</H2>** tags, your heading will be centered.]
- **<hr>** [The hr symbol placed in between sections of text makes a horizontal divider bar to separate material on your page. The line you see above emulates the line that will appear in your home page if you use the **<HR>** option.]



- ** Words between these tags will show as bold text**
- **<i> Words between these tags will show as italicized text</i>**
- **Link Tag: Name for the Web document you want to appear on the Web page **



- **Create an email link:** `<address>` e-mail me at: `` Put your email name with extension here also `</address>`
[The tags above will allow people to send you an e-mail by clicking on the link created when you type in the html tags above.]



- **Set Font Size and Color** other than the default settings:
** text**
[Experiment with color and font size to find what you like. It is easier to use an html editor to try out various sizes of font and colors.]



- **Indent quoted material:**
<blockquote> Insert material to be blocked
between these tags **</blockquote>**



- **Open a new window to display a linked site:**
`<a href="http://www.`
- **`google.com target="_blank">google`**
[The `target="blank"` tag inserted right after the Web address will take the user to the google, while still keeping the point-of-origin site page from which the link originated.]



- **Insert Graphics Files:**

**** [Graphics usually have a **.jpg or .gif** file extension. Insert the name of your image file between the quotation marks to replace the "picture.jpg" filename. The file could have a .gif extension. The "ALIGN =left" tag signifies the image is to be aligned to the left of the web page. You can choose left, right, center.]



- **Emphasis tag:** It is used to renders as emphasized tex
- ` Statements... `
- **Small (text) tag:** It is used to set the small font size of the content.**Syntax:**
- `<small> Statements... </small>`
- **Deleted text tag:** It is used to represent as deleted text. It cross the text content.**Syntax:**
- `<strike> Statements... </strike>`



- `Example`





- Some Mathematical Symbols Supported by HTML
- CharNumber Entity Description
- \forall `∀` `∀` FOR ALL
- ∂ `∂` `∂` PARTIAL DIFFERENTIAL
- \exists `∃` `∃` THERE EXISTS
- \emptyset `∅` `∅` EMPTY SETS
- ∇ `∇` `∇` NABLA
- \in `∈` `∈` ELEMENT OF
- \notin `∉` `∉` NOT AN ELEMENT OF
- \ni `∋` `∋` CONTAINS AS MEMBER
- \prod `∏` `∏` N-ARY PRODUCT
- \sum `∑` `∑` N-ARY SUMMATION



- | Char | Number | Entity | Description |
|------|--------|-----------|------------------------------|
| Α | Α | Α | GREEK CAPITAL LETTER ALPHA |
| Β | Β | Β | GREEK CAPITAL LETTER BETA |
| Γ | Γ | Γ | GREEK CAPITAL LETTER GAMMA |
| Δ | Δ | Δ | GREEK CAPITAL LETTER DELTA |
| Ε | Ε | Ε | GREEK CAPITAL LETTER EPSILON |
| Ζ | Ζ | Ζ | GREEK CAPITAL LETTER ZETA |



Char	Number	Entity	Description
•	©	© ©	COPYRIGHT SIGN
•	®	® ®	REGISTERED SIGN
•	€	€	€ EURO SIGN
•	™	™	™ TRADEMARK
•	←	←	← LEFTWARDS ARROW
•	↑	↑	↑ UPWARDS ARROW
•	→	→	→ RIGHTWARDS ARROW
•	↓	↓	↓ DOWNWARDS ARROW
•	♠	♠	♠ BLACK SPADE SUIT
•	♣	♣	♣ BLACK CLUB SUIT
•	♥	♥	♥ BLACK HEART SUIT
•	♦	♦	♦ BLACK DIAMOND SUIT



Style

link



- <https://www.htmlpad.net/download.php>

HTML | style attribute



- Styles in HTML are basically rules that describe how a document will be presented in a browser.
- There are 3 ways of implementing style in HTML :
- **Internal** : In this method, the style element is used inside the <head> element of the document.
- **Inline Style** : In this method, the style attribute is used inside the HTML start tag.
- **External Style Sheet** : In this method the <link> element is used to point to an external CSS file.

Internal Style



- Internal style sheet has two type of attribute
- 1)Type
- 2)Media
- Media attribute plays important roll in style. We can decide how the display will appear when open in small device, printing document etc.



```
<!doctype html>
<html>
<head>
<title> Style Sheet </title>
<style type = "text/css">
h1{color:red;font-size : 15px;}
p{color:red;font-size : 35px;} /* background-color: 009900;      */
</style>

</head>
<body>
<h1> hello this my first website !!! </h1>
<p> Styles in HTML are basically rules that describe how a document will be presented
in a browser. There are 3 ways of implementing style in HTML </p>
</body>
</html>
```

```
<!doctype html>
```



```
<html>
```

```
<head>
```

```
<title> Style Sheet </title>
```

```
<style type = "text/css">
```

```
h1{color:red;font-size : 15px;}
```

```
p{color:red;font-size : 35px;}</style>
```

```
<style type = "text/css" media ="print">
```

```
h1{color:green;font-size : 45px;}
```

```
p{color:green;font-size : 55px;}</style>
```

```
</head>
```

```
<body>
```

```
<h1> hello this my first website !!! </h1>
```

```
<p> Styles in HTML are basically rules  
that describe how a document will be  
presented in a browser. There are 3 ways of  
implementing style in HTML </p>
```

```
</body>
```

```
</html>
```



```
<!doctype html>
<html>
<head>
<title> Style Sheet </title>
<style type = "text/css">
h1{color:red;font-size : 15px;}
p{color:red;font-size : 35px;}</style>
```

```
<style type = "text/css" media ="screen and ( max-width:1000px) ">
h1{color:yellow;
background-color:green;
font-size : 55px;}
p{color:blue;font-size : 35px;}</style>
```

```
</head>
<body>
<h1> hello this my first website !!! </h1>
<p> Styles in HTML are basically rules that describe how a document will be presented in a browser. There are 3
ways of implementing style in HTML </p>
</body>
</html>
```

Border



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p.normal {`
- `border: 2px solid red;`
- `}`
- `p.round1 {`
- `border: 2px solid red;`
- `border-radius: 5px;`
- `}`
- `p.round2 {`
- `border: 2px solid red;`
- `border-radius: 8px;`
- `}`
- `p.round3 {`
- `border: 2px solid red;`
- `border-radius: 12px;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h2>The border-radius Property</h2>`
- `<p>This property is used to add rounded borders to an element:</p>`
- `<p class="normal">Normal border</p>`
- `<p class="round1">Round border</p>`
- `<p class="round2">Rounder border</p>`
- `<p class="round3">Roundest border</p>`
- `</body>`
- `</html>`



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p.dotted {border-style: dotted;}`
- `p.dashed {border-style: dashed;}`
- `p.solid {border-style: solid;}`
- `p.double {border-style: double;}`
- `p.groove {border-style: groove;}`
- `p.ridge {border-style: ridge;}`
- `p.inset {border-style: inset;}`
- `p.outset {border-style: outset;}`
- `p.none {border-style: none;}`
- `p.hidden {border-style: hidden;}`
- `p.mix {border-style: dotted dashed solid double;}`
- `</style>`
- `</head>`
- `<body>`
- `<h2>The border-style Property</h2>`
- `<p>This property specifies what kind of border to display:</p>`
- `<p class="dotted">A dotted border.</p>`
- `<p class="dashed">A dashed border.</p>`
- `<p class="solid">A solid border.</p>`
- `<p class="double">A double border.</p>`
- `<p class="groove">A groove border.</p>`
- `<p class="ridge">A ridge border.</p>`
- `<p class="inset">An inset border.</p>`
- `<p class="outset">An outset border.</p>`
- `<p class="none">No border.</p>`
- `<p class="hidden">A hidden border.</p>`
- `<p class="mix">A mixed border.</p>`
- `</body>`
- `</html>`



- The div tag is known as Division tag. The div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar, etc). Div tag has both open(<div>) and closing (</div>) tag and it is mandatory to close the tag. The Div is the most usable tag in web development because it helps us to separate out data in the web page and we can create a particular section for particular data or function in the web pages.



- Creating Web Layout using Div Tag
- The div tag is a container tag inside div tag we can put more than one HTML element and can group together and can apply CSS for them.
- div tag can be used for creating a layout of web we can also create web layout using tables tag but table tags are very complex to modify the layout
- The div tag is very flexible in creating web layouts and easy to modify.



```
<html>
<head>
<title>Div Exercise</title>
<style type=text/css>
    .leftdiv
    {
        float: left;
    }
    .middlediv
    {
        float: left;
        background-color:gray
    }
    .rightdiv
    {
        float: left;
    }
    div{
        padding : 1%;
        color: white;
        background-color: 009900;
        width: 30%;
        border: solid black;
    }
</style>
</head>
<body>
<div class="leftdiv">
    <h1>My Self</h1>
    <p>Hello every one this is sushant srivastava . I am a web developer and also a hacker. I am working in IIT Kanpur as a ASSISTANT prof.
</p>
    <h2>My Qualifications</h2>
    <p>B.Sc ,PGDCA,A+,N+,CCNA.MCSA,MCSE,CEKH,MCA,.
</p>
</div>
<div class="middlediv">
    <h1>tips</h1>
    <p>How many times were you frustrated while looking out
    for a good collection of programming/algorithm/interview
    questions? What did you expect and what did you get?
    This portal has been created to provide well written,
    well thought and well-explained solutions for selected
    questions.
    </p>
    <h2>Sushant Srivasatva</h2>
    <p>GCET is an entrance test for the extensive classroom
    programme by Sushant Srivasatva to build and enhance Data
    Structures and Algorithm concepts.
    He has 7 years of
    teaching experience and 6 years of industry experience.
    </p>
</div>
<div class="rightdiv">
    <h1>Sushant Srivasatva</h1>
    <p>How many times were you frustrated while looking out
    for a good collection of programming/algorithm/interview
    questions? What did you expect and what did you get?
    This portal has been created to provide well written,
    well thought and well-explained solutions for selected
    questions.
    </p>
    <h2>Sushant Srivasatva</h2>
    <p>How many times were you frustrated while looking out
```

opacity



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `div {`
- `background-color: green;`
- `}`
- `div.first {`
- `opacity: 0.1;`
- `}`
- `div.second {`
- `opacity: 0.3;`
- `}`
- `div.third {`
- `opacity: 0.6;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>Transparent Box</h1>`
- `<p>When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:</p>`
- `<div class="first">`
- `<h1>opacity 0.1</h1>`
- `</div>`
- `<div class="second">`
- `<h1>opacity 0.3</h1>`
- `</div>`
- `<div class="third">`
- `<h1>opacity 0.6</h1>`
- `</div>`
- `<div>`
- `<h1>opacity 1 (default)</h1>`
- `</div>`
- `</body>`
- `</html>`

Border



```
• <!DOCTYPE html>
• <html>
• <head>
• <style>
• p.one {
•   border-style: solid;
•   border-width: 5px;
• }

• p.two {
•   border-style: solid;
•   border-width: medium;
• }

• p.three {
•   border-style: dotted;
•   border-width: 2px;
• }

• p.four {
•   border-style: dotted;
•   border-width: thick;
• }

• p.five {
•   border-style: double;
•   border-width: 15px;
• }

• p.six {
•   border-style: double;
•   border-width: thick;
• }
• </style>
• </head>
• <body>

• <h2>The border-width Property</h2>
• <p>This property specifies the width of the four borders:</p>

• <p class="one">Some text.</p>
• <p class="two">Some text.</p>
• <p class="three">Some text.</p>
• <p class="four">Some text.</p>
• <p class="five">Some text.</p>
• <p class="six">Some text.</p>

• <p><b>Note:</b> The "border-width" property does not work if it is used alone.
• Always specify the "border-style" property to set the borders first.</p>

• </body>
• </html>
```

Different Sides

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p {`
- `border-top-style: dotted;`
- `border-right-style: solid;`
- `border-bottom-style: dotted;`
- `border-left-style: solid;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h2>Individual Border Sides</h2>`
- `<p>2 different border styles.</p>`
- `</body>`
- `</html>`



Inline Style



- Usually, CSS is written in a separate CSS file (with file extension .css) or in a <style> tag inside of the <head> tag, but there is a third place which is also valid. The third place you can write CSS is inside of an HTML tag, using the style attribute. When CSS is written using the style attribute, it's called an “inline style”. In general, this is not considered a best practice. However, there are times when inline styles are the right (or only) choice.
- `<tagname style="property:value;">`



- **WHEN TO USE INLINE STYLES**
- Professional web developers do not use inline styles often, but there are times when they are important to understand or necessary to use. Here are a few places you may see inline styles:
- HTML e-mail
- Older websites
- `<p style="color: blue; font-size: 46px;">`



- `<!DOCTYPE html>`

`<html>`

`<body>`

`<h1 style="color:blue;text-align:center;">This is a heading</h1>`

`<p style="color:red;">This is a paragraph.</p>`

`</body>`

`</html>`

External CSS.



- The **external** style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file. ... `<link rel="stylesheet" type="text/css" href="mystyle. css">`



Player

Audio Player



- `<!DOCTYPE html>`
- `<html lang="en">`
- `<head>`
- `<title>Example of HTML audio Tag with Alternate Sources</title>`
- `</head>`
- `<body>`
- `<audio controls="controls" source src="audio.mp3" type="audio/mpeg">`
- Your browser does not support the HTML5 audio element.
- `</audio>`
- `</body>`
- `</html>`

Video Player

- `<video controls width="250">`

- `<source src="vedio.mp4"`
• `type="video/mp4">`

- Sorry, your browser doesn't support embedded videos.

- `</video>`

Add Video From You Tube



- `<iframe width="560" height="315" src="https://www.youtube.com/embed/NL_qnNd-4oU" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>`



- Pseudo-Elements



- CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows –
- `selector: pseudo-element {property: value}`

Sr.No.	Value & Description
1	:first-line Use this element to add special styles to the first line of the text in a selector.
2	:first-letter Use this element to add special style to the first letter of the text in a selector.
3	:before Use this element to insert some content before an element.
4	:after Use this element to insert some content after an element.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `p:first-line { text-decoration: underline; }`
- `p.noline:first-line { text-decoration: none; }`
- `</style>`
- `</head>`
- `<body>`
- `<p class = "noline">`
- This line would not have any underline because this belongs to nline class.
- `</p>`
- `<p>`
- The first line of this paragraph will be underlined as defined in the
- CSS rule above. Rest of the lines in this paragraph will remain normal.
- This example shows how to use :first-line pseduo element to give effect
- to the first line of any HTML element.
- `</p>`
- `</body>`
- `</html>`



- The :first-letter pseudo-element
- The following example demonstrates how to use the :first-letter element to add special effects to the first letter of elements in the document.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `p:first-letter { font-size: 5em; }`
- `p.normal:first-letter { font-size: 10px; }`
- `</style>`
- `</head>`
- `<body>`
- `<p class = "normal">`
- First character of this paragraph will be normal and will have font size 10 px;
- `</p>`
- `<p>`
- The first character of this paragraph will be 5em big as defined in the
- CSS rule above. Rest of the characters in this paragraph will remain
- normal. This example shows how to use `:first-letter` pseduo element
- to give effect to the first characters of any HTML element.
- `</p>`
- `</body>`
- `</html>`



- The :before pseudo-element
- The following example demonstrates how to use the :before element to add some content before any element.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `p:before {`
- `content: url(/images/200.gif)`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<p> This line will be preceded by a bullet.</p>`
- `</body>`
- `</html>`



- The :after pseudo-element
- The following example demonstrates how to use the :after element to add some content after any element



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `p:after {`
- `content: url(/images/bullet.gif)`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<p> This line will be succeeded by a bullet.</p>`
- `</body>`
- `</html>`



Pseudo Classes



- CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows –
- `selector:pseudo-class {property: value}`
- CSS classes can also be used with pseudo-classes –
- `selector.class:pseudo-class {property: value}`

The most commonly used pseudo-classes are as follows –



Sr.No.	Value & Description
1	:link Use this class to add special style to an unvisited link.
2	:visited Use this class to add special style to a visited link.
3	:hover Use this class to add special style to an element when you mouse over it.
4	:active Use this class to add special style to an active element.



5

:focus

Use this class to add special style to an element while the element has focus.



- While defining pseudo-classes in a `<style>...</style>` block, following points should be noted –
- `a:hover` MUST come after `a:link` and `a:visited` in the CSS definition in order to be effective.
- `a:active` MUST come after `a:hover` in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `a:link {color:#000000}`
- `</style>`
- `</head>`
- `<body>`
- `Black Link`
- `</body>`
- `</html>`

The :visited pseudo-class



- The following is the example which demonstrates how to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `a:visited {color: #006600}`
- `</style>`
- `</head>`
- `<body>`
- `Click this link`
- `</body>`
- `</html>`

The :hover pseudo-class



- The following example demonstrates how to use the *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `a:hover {color: #FFCC00}`
- `</style>`
- `</head>`
- `<body>`
- `Bring Mouse Here`
- `</body>`
- `</html>`

The :active pseudo-class



- The following example demonstrates how to use the *:active* class to change the color of active links. Possible values could be any color name in any valid format.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `a:active {color: #FFooCC}`
- `</style>`
- `</head>`
- `<body>`
- `Click This Link`
- `</body>`
- `</html>`

The :focus pseudo-class



- The following example demonstrates how to use the *:focus* class to change the color of focused links. Possible values could be any color name in any valid format.



- `<html>`
- `<head>`
- `<style type = "text/css">`
- `a:focus {color: #0000FF}`
- `</style>`
- `</head>`
- `<body>`
- `Click this Link`
- `</body>`
- `</html>`

web safe fonts css



- The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems. If the browser does not support the first font, it tries the next font.



font-family

Georgia, serif

Palatino Linotype", "Book Antiqua",
Palatino, serif

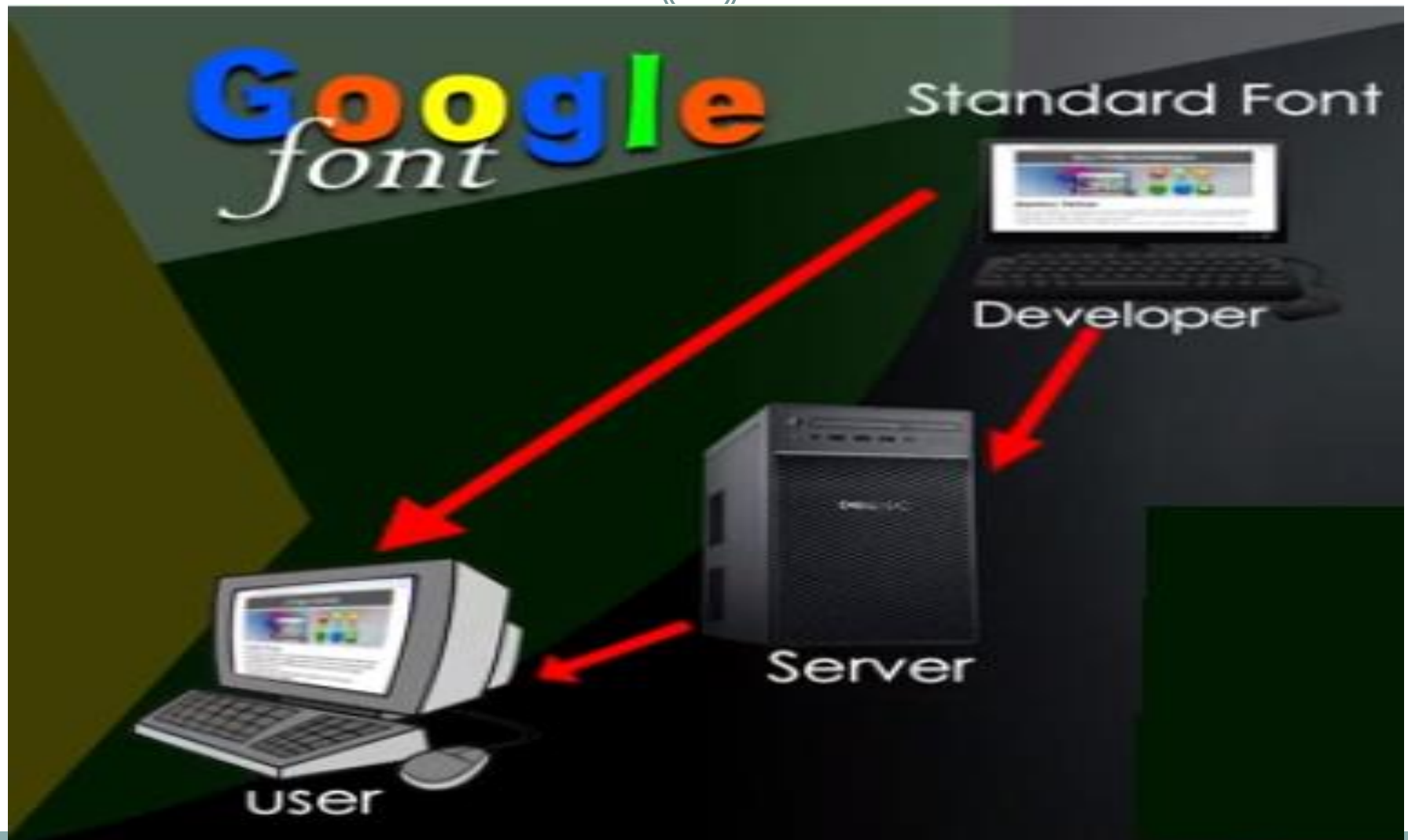
"Times New Roman", Times, serif

Sans-Serif Fonts

Georgia, serif

"Palatino Linotype", "Book Antiqua",
Palatino, serif

"Times New Roman", Times, serif



TABLE



- The `<table>` tag defines an HTML table.
- An HTML table consists of one `<table>` element and one or more [`<tr>`](#), [`<th>`](#), and [`<td>`](#) elements.
- The `<tr>` element defines a table row, the `<th>` element defines a table header, and the `<td>` element defines a table cell.



- `<table>`
- `<tr>`
- `<th>Month</th>`
- `<th>Savings</th>`
- `</tr>`
- `<tr>`
- `<td>January</td>`
- `<td>$100</td>`
- `</tr>`
- `</table>`



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `table, th, td {`
- `border: 1px solid black;`
- `}`
- `</style>`
- `</head>`
- `<body>`

- `<h1>Right-align a Table</h1>`

- `<table style="float:right">`
- `<tr>`
- `<th>Month</th>`
- `<th>Savings</th>`
- `</tr>`
- `<tr>`
- `<td>January</td>`
- `<td>$100</td>`
- `</tr>`
- `<tr>`
- `<td>February</td>`
- `<td>$80</td>`
- `</tr>`
- `</table>`
- `<p>This is some text to be wrapped around the table. This is some text to be wrapped around the table. This is some text to be wrapped around the table.</p>`

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



- <!DOCTYPE html>
- <html>
- <head>
- <style>
- table, th, td {
- border: 1px solid black;
- }
- table.center {
- margin-left: 10PX;
- margin-right: auto;
- }
- </style>
- </head>
- <body>
- <h1>Center a Table</h1>
- <p>To center a table, set left and right margin to auto:</p>
- <table class="center">
- <tr>
- <th>Month</th>
- <th>Savings</th>
- </tr>
- <tr>
- <td>January</td>
- <td>\$100</td>
- </tr>
- <tr>
- <td>February</td>
- <td>\$80</td>
- </tr>
- </table>
- </body>
- </html>



- <!DOCTYPE html>
- <html>
- <head>
- <style>
- table, th, td {
- border: 1px solid black;
- }
- </style>
- </head>
- <body>

- <h1>Add Background Color to a Table</h1>

- <table style="background-color:#00FF00">
- <tr>
- <th>Month</th>
- <th>Savings</th>
- </tr>
- <tr>
- <td>January</td>
- <td>\$100</td>
- </tr>
- <tr>
- <td>February</td>
- <td>\$80</td>
- </tr>
- </table>

- </body>
- </html>



- <!DOCTYPE html>
- <html>
- <head>
- <style>
- table, th, td {
- border: 1px solid black;
- }
- th, td {
- padding: 10px;
- }
- </style>
- </head>
- <body>
- <h1>Add Padding to a Table</h1>
- <table>
- <tr>
- <th>Month</th>
- <th>Savings</th>
- </tr>
- <tr>
- <td>January</td>
- <td>\$100</td>
- </tr>
- <tr>
- <td>February</td>
- <td>\$80</td>
- </tr>
- </table>
- </body>
- </html>



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `table, th, td {`
- `border: 1px solid black;`
- `}`
- `</style>`
- `</head>`
- `<body>`

- `<h1>The caption element</h1>`

- `<table>`
- `<caption>Monthly savings</caption>`
- `<tr>`
- `<th>Month</th>`
- `<th>Savings</th>`
- `</tr>`
- `<tr>`
- `<td>January</td>`
- `<td>$100</td>`
- `</tr>`
- `<tr>`
- `<td>February</td>`
- `<td>$50</td>`
- `</tr>`
- `</table>`

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



- <!DOCTYPE html>
- <html>
- <head>
- <style>
- table {
- display: table;
- border-collapse: separate;
- border-spacing: 2px;
- border-color: gray;
- }
- </style>
- </head>
- <body>
- <table>
- <tr>
- <th>Month</th>
- <th>Savings</th>
- </tr>
- <tr>
- <td>January</td>
- <td>\$100</td>
- </tr>
- <tr>
- <td>February</td>
- <td>\$80</td>
- </tr>
- </table>
- </body>
- </html>

HTML Tag



- The HTML tag is used for specifying an unordered list, which groups a collection of items having no numerical order. When changing the order of list items, the meaning does not change. Usually, the items of an unordered list are displayed with a bullet. It can be of different forms such as a circle, a dot, or a square.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>Title of the document</title>`
- `</head>`
- `<body>`
- ``
- `List item`
- `List item`
- `List item`
- ``
- `</body>`
- `</html>`



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>Title of the document</title>`
- `</head>`
- `<body>`
- `<ul type="circle">`
- `List item `
- `List item`
- `List item`
- ``
- `<ul type="square">`
- `List item`
- `List item`
- `List item`
- ``
- `</body>`
- `</html>`



- <!DOCTYPE html>
- <html>
- <head>
- <title>Title of the document</title>
- </head>
- <body>
- <h2>Examples of unordered lists:</h2>
- <ul style="list-style-type: square">
- Cold Drinks
- Hot Drinks
- Ice-Creams
-
- <ul style="list-style-type: disc">
- Coca-Cola
- Fanta
- Ice Tea
-
- <ul style="list-style-type: circle">
- Coca-Cola
- Fanta
- Ice Tea
-
- </body>
- </html>

HTML



- The **HTML element** represents an ordered list of items — typically rendered as a numbered list.



- ``
- `Fee`
- `Fi`
- `Fo`
- `Fum`
- ``

Using Roman Numeral type



- `<ol type="i">`
- `Introduction`
- `List of Greivances`
- `Conclusion`
- ``

Using the start attribute



- `<ol start="4">`
- `Speedwalk Stu`
- `Saunterin' Sam`
- `Slowpoke Rodriguez`
- ``

Nesting lists



- ``
- `first item`
- `second item <!-- closing tag not here! -->`
- ``
- `second item first subitem`
- `second item second subitem`
- `second item third subitem`
- ``
- ` <!-- Here's the closing tag -->`
- `third item`
- ``

HTML 5 | <header> Tag



- The <header> tag in HTML is used to define the header for a document or a section.
- The header tag contains information related to the title and heading of the related content.
- The <header> element is intended to usually contain the section's heading (an h1-h6 element or an <hgroup> element), but this is not required.
- The <header> element can also be used to wrap a section's table of contents, a search form, or any relevant logos.
- The <header> tag is a new tag in HTML5 and it requires a starting tag as well as an end tag.
- There can be several <header> elements in one document.
- A <header> tag cannot be placed within a <footer>, <address> or another <header> element.
- **Syntax:**
- **<header> ...</header>**

nav



- One HTML document may contain several `<nav>` tags, for example, one for site navigation and one for intra-page navigation.
- Note that not all links in the HTML document are placed inside the `<nav>` element, it includes major navigation blocks. The `<nav>` tag can be placed for defining links in the footer of the website, but the [`<footer>`](#) tag is usually used in such cases.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>Title of the document</title>`
- `</head>`
- `<body>`
- `<header>`
- `<h1>Programming Courses</h1>`
- `</header>`
- `<nav>`
- `HTML |`
- `CSS |`
- `JavaScript |`
- `PHP |`
- `</nav>`
- `<h2>Welcome to W3Docs!</h2>`
- `</body>`
- `</html>`



```
• <!DOCTYPE html>
• <html>
• <head>
• <style>
•   nav {
•     display: flex;
•     flex-wrap: wrap;
•   }
•   nav a {
•
•     padding: 15px 25px;
•     text-align: center;
•     background-color: rgb(189, 185, 185);
•
•
•     font-family: sans-serif;
•   }
•   nav a:hover {
•     background-color: #777777;
•     color: #ffffff;
•   }
• </style>
• </head>
• <body>
• <h1>Example of the HTML nav tag:</h1>
• <nav>
•   <a href="https://www.w3docs.com/">Home</a>
•   <a href="https://www.w3docs.com/quiz/">Quizzes</a>
•   <a href="https://www.w3docs.com/snippets">Snippets</a>
•   <a href="https://www.w3docs.com/tool/">Tools</a>
•   <a href="https://www.w3docs.com/string-functions/">String Functions</a>
• </nav>
• </body>
• </html>
```

HTML Section Tag



- The HTML `<section>` tag is used to define sections in a document. When you put your content on a web page, it may contains many chapters, headers, footers, or other sections on a web page that is why HTML `<section>` tag is used.
- HTML `<section>` is a new tag introduced in HTML5.



- <!DOCTYPE>
- <html>
- <head>
- <style>
- section{
- border:1px solid pink;
- padding:15px;
- margin:10px;
- }
- </style>
- </head>
- <body>
- <h2> Indian Leader</h2>
- <section>
- <h3> Jawaharlal Nehru </h3>
- <p> Jawaharlal Nehru was the first Prime Minister of India and a central figure in
- Indian politics for much of the 20th century. He emerged as the paramount leader of the Indian
- independence movement under the tutelage of Mahatma Gandhi. -Source Wikipedia </p>
- </section>
- <section>
- <h3>Subhas Chandra Bose </h3>
- <p>Subhas Chandra Bose was an Indian nationalist whose attempt during World War II to rid India of
- British rule with the help of Nazi Germany and Japan left a troubled legacy.
- The honorific Netaji (Hindustani language: "Respected Leader"), first applied to Bose in Germany,
- by the Indian soldiers of the Indische Legion and by the German and Indian officials
- in the Special Bureau for India in Berlin, in early 1942, is now used widely throughout India. -source Wikipedia</p>
- </section>
- </body>
- </html>
-



- The **<article>** tag is one of the new sectioning element in HTML5. The HTML **<article>** tag is used to represent an article. More specifically, the content within the **<article>** tag is independent from the other content of the site (even though it can be related).
- In other words, The article element represents a component of a page that consists of self-contained composition in a document, page or a site. For Ex. in syndication.
Potential source for Article Element are :
 - A magazine/newspaper article
 - A blog entry
 - A forum post
 - A user submitted comment



- `<!DOCTYPE html>`
- `<html>`
-
- `<head>`
- `<title>Demo of article</title>`
-
- `</head>`
-
- `<body>`
- `<article style="width: 300px; border: 2px solid gray;`
- `padding: 10px; border-radius: 10px;`
- `margin:5px;">`
- `<h1> content</h1>`
- `<content </p>`
- `</article>`
- `</body>`
-
- `</html>`

Supported Browser:



- The browser supported by <article> tag are listed below :
- Google Chrome 6.0
- Internet Explorer 9.0
- Firefox 4.0
- Opera 11.1
- Safari 5.0

Semantic Elements?



- HTML was originally created as a markup language to describe documents on the early internet. As the internet grew and was adopted by more people, its needs changed.
- Where the internet was originally intended for sharing scientific documents, now people wanted to share other things as well. Very quickly, people started wanting to make the web look nicer.
- Because the web was not initially built to be designed, programmers used different hacks to get things laid out in different ways. Rather than using the `<table></table>` to describe information using a table, programmers would use them to position other elements on a page.



- The semantic elements added in HTML5 are:
- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



- Why use semantic elements?
- To look at the benefits of semantic elements, here are two pieces of HTML code. This first block of code uses semantic elements:



- `<header></header>`
- `<section>`
- `<article>`
- `<figure>`
- ``
- `<figcaption></figcaption>`
- `</figure>`
- `</article>`
- `</section>`
- `<footer></footer>`



- `<div id="header"></div>`
- `<div class="section">`
- `<div class="article">`
- `<div class="figure">`
- ``
- `<div class="figcaption"></div>`
- `</div>`
- `</div>`
- `</div>`
- `<div id="footer"></div>`

HTML 5 | <header> Tag



- The <header> tag in HTML is used to define the header for a document or a section.
- The header tag contains information related to the title and heading of the related content.
- The <header> element is intended to usually contain the section's heading (an h1-h6 element or an <hgroup> element), but this is not required.
- The <header> element can also be used to wrap a section's table of contents, a search form, or any relevant logos.
- The <header> tag is a new tag in HTML5 and it requires a starting tag as well as an end tag.
- There can be several <header> elements in one document.
- A <header> tag cannot be placed within a <footer>, <address> or another <header> element.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>Header Tag</title>`
- `</head>`
- `<body>`
- `<article>`
- `<header>`
- `<h1>This is the heading.</h1>`
- `<h4>This is the sub-heading.</h4>`
- `<p>This is the metadata.</p>`
- `</header>`
- `</article>`
- `</body>`
- `</html>`

HTML | `<nav>` Tag



- The `<nav>` tag is used to declaring the navigational section in HTML documents. Websites typically have sections dedicated to navigational links, which enables user to navigate the site. These links can be placed inside a nav tag.
- In other words, nav element represents section of page whose purpose is to provide navigational links, either in current document or to other document. The links in nav element may points to other webpages or to different sections of same webpage. It is semantic element. Common examples of nav element are menus, tables, contents, and indexes.
- Syntax:
 - `<nav>`
 - Links...
 - `</nav>`



- <!DOCTYPE html>
- <html>
- <head>
- <title>nav tag</title>
- <style>
- .g {
- font-size:40px;
- color:#090;
- font-weight:bold;
- text-align:center;
- }
- .nav_tag {
- text-align:center;
- margin:30px 0;
- }
- </style>
- </head>
- <body>
- <div class = "g">sushant srivastava</div>
- <div class = "nav_tag">Nav Tag Example</div>
- <nav>
- Home |
- Interview |
- Languages |
- Data Structure |
- Algorithm
- </nav>
- </body>
- </html>



```
• <!DOCTYPE html>
• <html>
• <head>
• <title>nav tag</title>
• <style>
•   nav {
•     border:1px;
•     background-color:#090;
•     color:white;
•     padding:6px;
•   }
•   a {
•     text-decoration:none;
•     color:white;
•     font-size:20px;
•   }
•   .g{
•     font-size:40px;
•     color:#090;
•     font-weight:bold;
•     text-align:center;
•   }
•   .nav_tag {
•     text-align:center;
•     margin:30px 0;
•   }
• </style>
• </head>
• <body>
• <div class = "g">Google</div>
• <div class = "nav_tag">Nav Tag Example</div>
• <nav>
•   <a href =
• "https://www.google.org/">
•     Home</a> |
•   <a href =
• "https://www.google.org/company-interview-corner/">
•     Interview</a> |
•   <a href =
• "https://www.google.org/gate-cs-notes-gq/">
•     Gate</a> |
•   <a href =
• "https://www.google.org/data-structures/">
•     Data Structure</a> |
•   <a href =
• "https://www.google.org/fundamentals-of-algorithms/">
•     Algorithm</a>
• </nav>
• </body>
• </html>
```



- Section tag defines the section of documents such as chapters, headers, footers or any other sections. The section tag divides the content into section and subsections. The section tag is used when requirements of two headers or footers or any other section of documents needed. Section tag grouped the generic block of related contents. The main advantage of the section tag is, it is a semantic element, which describes its meaning to both browser and developer.



- `<h1>How </h1>`
- `<section>`
- `<h2>Introduction</h2>`
- `<p>People have been catching fish for food since before recorded history...</p>`
- `</section>`

- `<section>`
- `<h2>Equipment</h2>`
- `<p>The first thing you'll need is a fishing rod or pole that you find comfortable and is strong enough for the kind of fish you're expecting to land...</p>`
- `</section>`



- HTML5 | article tag
- The <article> tag is one of the new sectioning element in HTML5. The HTML <article> tag is used to represent an article. More specifically, the content within the <article> tag is independent from the other content of the site (even though it can be related).
- In other words, The article element represents a component of a page that consists of self-contained composition in a document, page or a site. For Ex. in syndication.
- Potential source for Article Element are :
 - A magazine/newspaper article
 - A blog entry
 - A forum post
 - A user submitted comment



- This tag is most often used in two contexts:
- On a page with a single piece of content, a single `<article>` element can be used to contain the main content and set it off from the rest of the page.
- On a page with multiple pieces of content (a blog index page, a search results page, a category page, a news feed), multiple `<article>` elements can be used to contain each individual piece of content.



- <!DOCTYPE html>
- <html>
-
- <head>
- <title>Demo of article</title>
-
- </head>
-
- <body>
- <article style="width: 300px; border: 2px solid gray;
- padding: 10px; border-radius: 10px;
- margin: 5px;">
- <img src=
- "imgpath"
- alt="" width="300" height="250"
- class="alignnone size-medium wp-image-560930" />
- <h1> Google</h1>
- <p>
- Sushant srivastava has worked
- with DE Shaw and Co. as a software developer and IIIT Noida
- as an assistant professor.</p>
- </article>
- </body>
-
- </html>



- HTML5 | `<aside>` Tag
- The `<aside>` tag is used to describe the main object of the web page in a shorter way like a highlighter. It basically identifies the content that is related to the primary content of the web page but does not constitute the main intent of the primary page. The `<aside>` tag contains mainly author information, links, related content and so on.
- `<aside>` Vs `<div>`: Both tags have same behavior with different meaning.
- `<div>`: It define or create division or section in the web page.
- `<aside>`: It does the same job by creating section or division but it contains only the content that is related to the main web page.
- The `<aside>` tag makes easy to design the page and it enhances clarity of html document. It let us easily recognise the main text and subordinate text. In both the time `<div>` and `<aside>` need CSS to specific design. The `<aside>` tag supports Global attributes and Event attributes in HTML.



- `<html>`
- `<head>`
- `<title>aside tag</title>`
- `<style>`
- `.g{`
- `font-size:30px;`
- `}`
- `h1 {`
- `font-size:40px;`
- `color:#090;`
- `font-weight:bold;`
- `}`
- `p {`
- `font-size:20px;`
- `margin:20px 0;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<div class = "g">Aside Tag Example</div>`
- `<aside>`
- `<h1>sushant srivastava </h1>`
- `<p>A computer science portal for geeks</p>`
- `</aside>`
- `</body>`
- `</html>`



- The `<footer>` tag in HTML is used to define a footer of HTML document. This section contains the footer information (author information, copyright information, carriers, etc). The footer tag is used within the body tag. The `<footer>` tag is new in the HTML5. The footer elements require a start tag as well as an end tag.



- <!DOCTYPE html>
- <html>
- <head>
- <title>HTML footer Tag</title>
- <style>
- a {
- font-size:25px;
- text-decoration:none;
- }
- p {
- font-size:25px;
- }
- </style>
- </head>
- <body>
- <footer>
- <nav>
- <p>
- <a href=
- "https://www.google.org/about/">About Us|
- <a href=
- "https://www.google.org/privacy-policy/">Privacy Policy|
- <a href=
- "https://www.google.org/careers/">Careers
- </p>
- </nav>
- <p>@google, Some rights reserved</p>
- </footer>
- </body>
- </html>



```
<!DOCTYPE html>
<html>
  <head>
    <title>footer tag</title>
    <style>
      .column {
        float: left;
        width: 27%;
        height: 300px;
      }
      p {
        font-size: 20px;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <footer>
      <div class="column">
        <p>Company</p>
        <ul style="list-style-type: disc">
          <li>About Us</li>
          <li>Careers</li>
          <li>Privacy Policy</li>
          <li>Contact Us</li>
        </ul>
      </div>
      <div class="column">
        <p>Learn</p>
        <ul>
          <li>Algorithms</li>
          <li>Data Structures</li>
          <li>Languages</li>
          <li>CS Subjects</li>
          <li>Video Tutorials</li>
        </ul>
      </div>
      <div class="column">
        <p>Practice</p>
        <ul>
          <li>Company-wise</li>
          <li>Topic-wise</li>
          <li>Contests</li>
          <li>Subjective Questions</li>
        </ul>
      </div>
    </footer>
  </body>
</html>
```



- HTML5 | `<details>` tag
- The `<details>` tag is used for the content/information which is initially hidden but could be displayed if the user wishes to see it. This tag is used to create an interactive widget which the user can open or close it. The content of the details tag is visible when open the set attributes.
- Syntax:
 - `<details>`
 - `<summary>` Text content `</summary>`
 - `<div>` Content . . . `>`
 - `</details>`



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>details tag</title>`
- `<style>`
- `summary {`
- `font-size:40px;`
- `color:#090;`
- `font-weight:bold;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<details>`
- `<summary>Google</summary>`
- `<p>A computer science portal for geeks</p>`
- `<div>It is a computer science portal where you`
- `can learn programming.</div>`
- `</details>`
- `</body>`
- `</html>`



- HTML5 | figure Tag
- The `<figure>` tag in HTML is used to add self-contained content like illustrations, diagrams, photos or codes listing in a document. It is related to main flow but it can be used in any position of a document and the figure goes with the flow of the document and if remove it then it should not affect the flow of the document. This tag is new in HTML5.
- Syntax:
- `<figure>` Image content... `</figure>`
- Attributes: It contains mostly two tags which are listed below:
- `img src`: This tag is used to add image source in the document.
- `figcaption`: This tag is use to set the caption to the image.



```
• <!DOCTYPE html>
• <html>
•   <head>
•     <title>figure tag</title>
•     <style>
•       body {
•         text-align:center;
•       }
•       h1 {
•         color:green;
•       }
•     </style>
•   </head>
•   <body>
•     <h1>Google</h1>
•     <h2><figure> Tag</h2>
•     <figure>
•       <img src=
• "https://media.google.org/wp-content/uploads/geeks-25.png"
•       alt="The Pulpit Rock" width="304" height="228">
•       <figcaption> logo</figcaption>
•     </figure>
•   </body>
• </html>
```

Image Link



- Images are not technically inserted into a web page; images are linked to web pages. The `` tag creates a holding space for the referenced image.
- The `` tag is empty, it contains attributes only, and does not have a closing tag.
- The `` tag has two required attributes:
- `src` - Specifies the path to the image
- `alt` - Specifies an alternate text for the image

Type Of Linking



- There are two types of linking present
- 1)Relative(Normal) Linking.
- 2)Absolute Linking.

Relative Linking



- `<html>`
- `<head>`
- `<title>`
- `</title>`
- `</head>`
- `<body>`
- ``
- `</body>`
- `</html>`

Absolute/Hot Linking



- `<html>`
- `<head>`
- `</head>`
- `<body>`
- `<img src`
= "<https://assets.pcmag.com/media/images/599652-hp-eliteone-1000-curved-aio-7.jpg?thumb=y>"
style="border:1px solid black; width:300px;
height:225px;">
- `</body>`
- `</html>`

Linking With In <a>



- `<html>`
- `<head>`
- `</head>`
- `<body>`
- ``
- `<img src`
`= "https://assets.pcmag.com/media/images/599652-hp-`
`eliteone-1000-curved-aio-7.jpg?thumb=y" style="border:1px`
`solid black; width:300px; height:225px;">`
- `</body>`
- `</html>`



- HTML | target Attribute
- The **HTML target Attribute** is used to specify where to open the linked document. It can be used on various elements such as:
- **Syntax:**
- `<element target="_blank|_self|_parent|_top|frameName">`**Attribute**
- **Values:**
- **_blank:** It opens the link in a new window.
- **_self:** It opens the linked document in the same frame.
- **_parent:** It opens the linked document in the parent frameset.
- **_top:** It opens the linked document in the full body of the window.
- **frameName:** It opens the linked document in the named frame.

Relative vs Absolute



- A relative link is any link that shows the current URL's relation to the linked page's URL. Instead of having the whole reference URL in the href tag, relative links only show the relative link paths, as their name suggests. It's the difference between a link that has the website's full address, and a link that only includes a file's specific location.
- For example, a relative link would only include `/images/puppies.gif` in the href tag. An absolute link, on the other hand, would include `https://www.example.com/image/puppies.gif` in it.
- Using relative links instead of absolute links when building a website won't make the site more appealing to search engines, which convert relative links into absolute links automatically. It doesn't really matter to search engines which ones get used.

Bookmark in HTML



- Create a Bookmark in HTML
- Bookmarks can be useful if a web page is very long.
- To create a bookmark - first create the bookmark, then add a link to it.
- When the link is clicked, the page will scroll down or up to the location with the bookmark.
- Example
- First, use the id attribute to create a bookmark:

BookMark



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<p>Jump to Chapter 4</p>`
- `<p>Jump to Chapter 10</p>`

- `<h2>Chapter 1</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 2</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 3</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2 id="C4">Chapter 4</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 5</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 6</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 7</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 8</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 9</h2>`
- `<p>This chapter explains ba bla bla</p>`

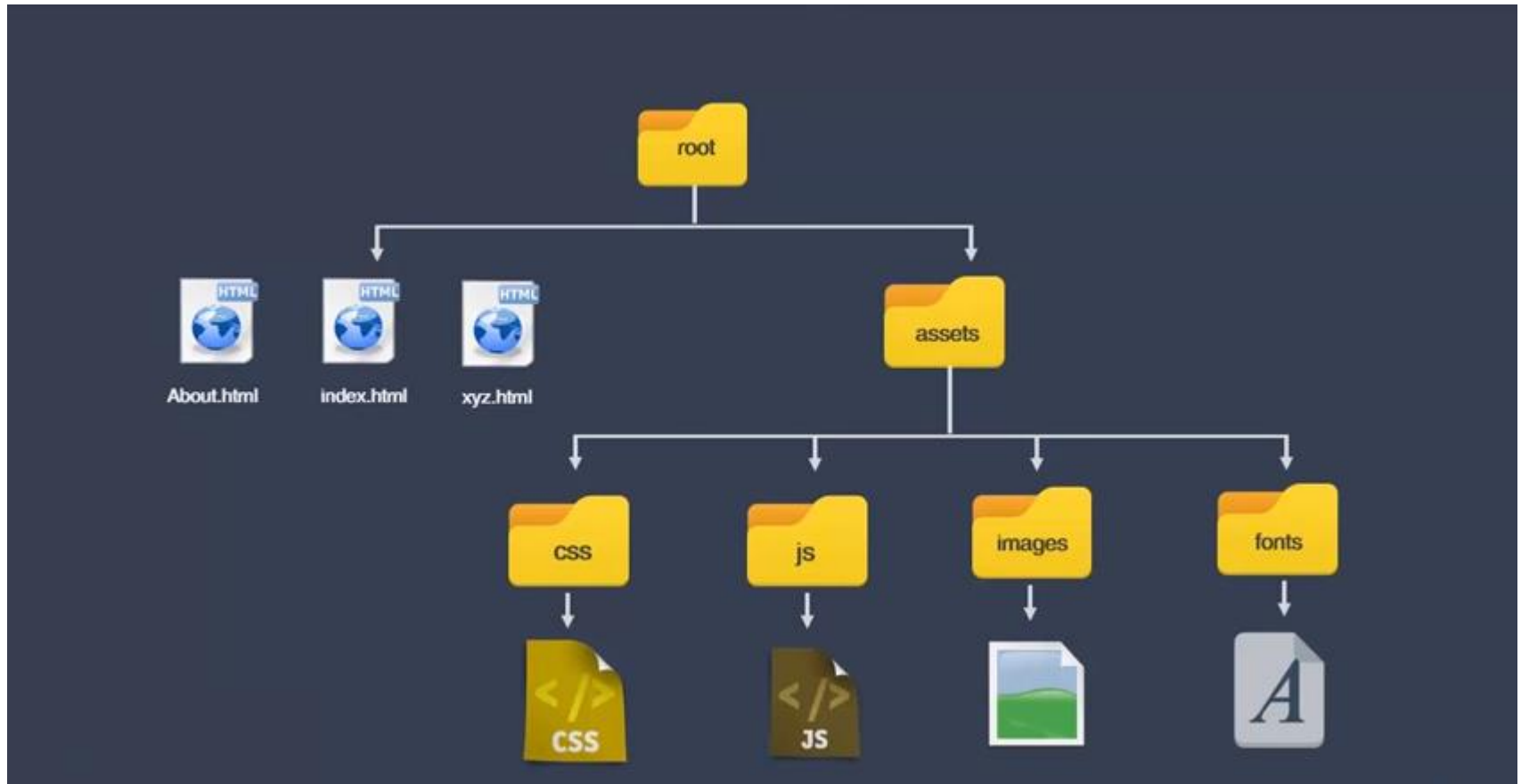
- `<h2 id="C10">Chapter 10</h2>`
- `<p>This chapter explains ba bla bla</p>`

- `<h2>Chapter 11</h2>`
- `<p>This chapter explains ba bla bla</p>`

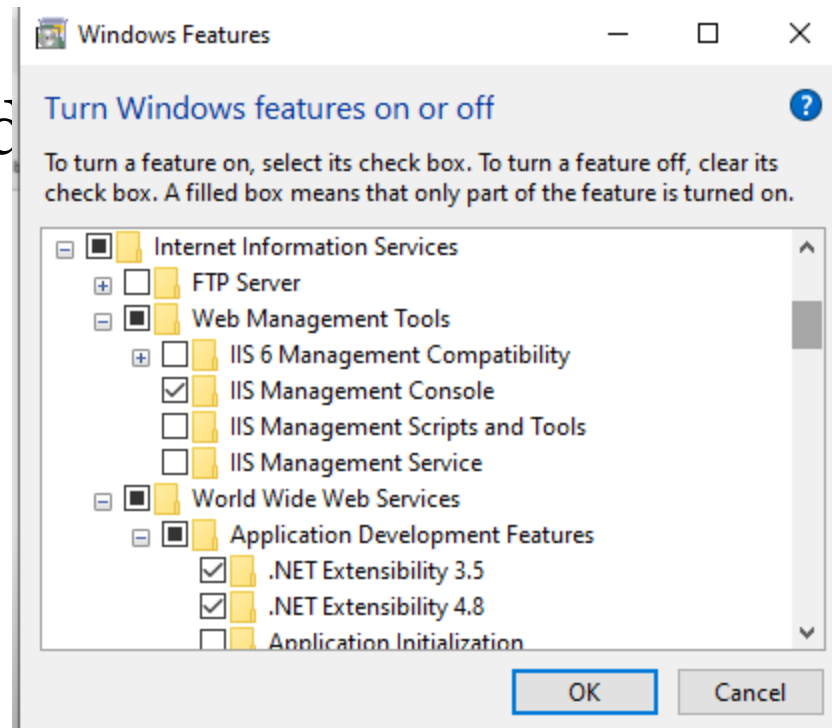
- `</body>`
- `</html>`

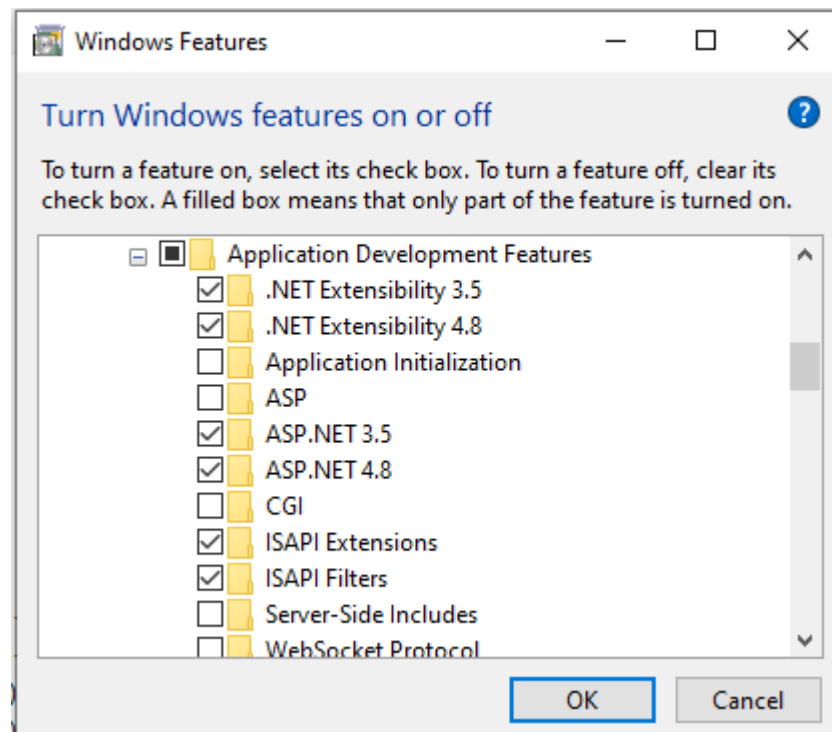


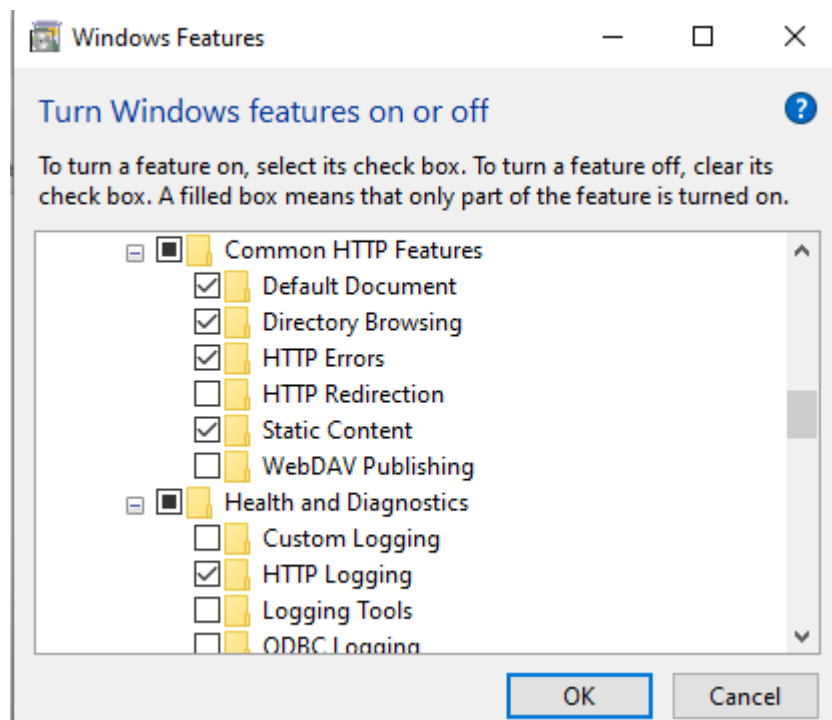
Folder Hierarchy

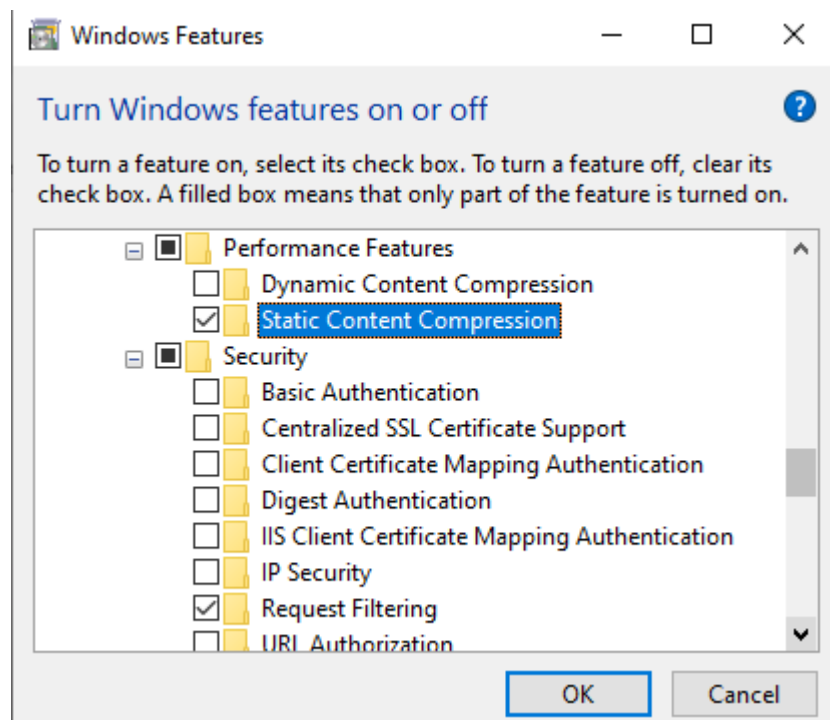


- Appwiz.cpl
- Trunon wind











- inetmgr

HTML action attribute



- The action attribute of `<form>` element defines the process to be performed on form when form is submitted, or it is a URI to process the form information.
- The action attribute value defines the web page where information proceed. It can be `.php`, `.jsp`, `.asp`, etc. or any URL where you want to process your form.



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The form element</h1>`

- `<form action="/action_page.html">`
- `<label for="fname">First name:</label>`
- `<input type="text" id="fname" name="fname">

`
- `<label for="lname">Last name:</label>`
- `<input type="text" id="lname" name="lname">

`
- `<input type="submit" value="Submit">`
- `</form>`

- `<p>Click the "Submit" button and the form-data will be sent to a page "</p>`

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

Get Method



- The form-data can be sent as URL variables (with `method="get"`) or as HTTP post transaction (with `method="post"`).

Appends form-data into the URL in
name/value pairs

The length of a URL is limited (about 3000
characters)

Never use GET to send sensitive data! (will
be visible in the URL)

Useful for form submissions where a user
wants to bookmark the result

GET is better for non-secure data, like
query strings in Google



- `<form action="/action_page.php" method="GET">`
- `<label for="fname">First name:</label>`
- `<input type="text" id="fname"`
`name="fname">

`
- `<label for="lname">Last name:</label>`
- `<input type="text" id="lname"`
`name="lname">

`
- `<input type="submit" value="Submit">`
- `</form>`

POST Method



- Appends form-data inside the body of the HTTP request (data is not shown in URL)
- Has no size limitations
- Form submissions with POST cannot be bookmarked



- `<form action="/action_page.php" method="post">`
- `<label for="fname">First name:</label>`
- `<input type="text" id="fname"`
`name="fname">

`
- `<label for="lname">Last name:</label>`
- `<input type="text" id="lname"`
`name="lname">

`
- `<input type="submit" value="Submit">`
- `</form>`

Input Type



- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`

The value Attribute



- The input value attribute specifies an initial value for an input field:
- `<form>`
 - `<label for="fname">First name:</label>
`
 - `<input type="text" id="fname" name="fname" value="SUSHANT">
`
 - `<label for="lname">Last name:</label>
`
 - `<input type="text" id="lname" name="lname" value="SRIVASTAVA">`
- `</form>`

The readonly Attribute



- The input readonly attribute specifies that an input field is read-only.
- A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).
- The value of a read-only input field will be sent when submitting the form!



- `<form>`
 `<label for="fname">First name:</label>
`
 `<input type="text" id="fname" name="fname" value="sushant" readonly>
`
 `<label for="lname">Last name:</label>
`
 `<input type="text" id="lname" name="lname" value="srivastava">`
 `</form>`

The disabled Attribute



- The input disabled attribute specifies that an input field should be disabled.
- A disabled input field is unusable and un-clickable.
- The value of a disabled input field will not be sent when submitting the form!



- `<form>`
 `<label for="fname">First name:</label>
`
 `<input type="text" id="fname" name="fname" value="Sushant" disabled>
`
 `<label for="lname">Last name:</label>
`
 `<input type="text" id="lname" name="lname" value="srivastava">`
 `</form>`

The size Attribute



- The input size attribute specifies the visible width, in characters, of an input field.
- The default value for size is 20.

```
<form>
  <label for="fname">First
name:</label><br>
  <input type="text" id="fname" name="fn
ame" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" s
ize="4">
</form>
```

The maxlength Attribute



- The input maxlength attribute specifies the maximum number of characters allowed in an input field.
- **Note:** When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.



- `<form>`
 `<label for="fname">First name:</label>
`
 `<input type="text" id="fname" name="fname" size`
 `= "50">
`
 `<label for="pin">PIN:</label>
`
 `<input type="text" id="pin" name="pin" maxlength`
 `= "4" size="4">`
 `</form>`



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The input min and max attributes</h1>`

- `<p>The min and max attributes specify the minimum and maximum values for an input element.</p>`

- `<form action="/action_page.php">`
- `<label for="datemax">Enter a date before 1980-01-01:</label>`
- `<input type="date" id="datemax" name="datemax" max="1979-12-31">

`

- `<label for="datemin">Enter a date after 2000-01-01:</label>`
- `<input type="date" id="datemin" name="datemin" min="2000-01-02">

`
- `<label for="quantity">Quantity (between 1 and 5):</label>`
- `<input type="number" id="quantity" name="quantity" min="1" max="5">

`

- `<input type="submit" value="Submit">`
- `</form>`

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

The required Attribute



- The input required attribute specifies that an input field must be filled out before submitting the form.
- The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The input required attribute</h1>`

- `<p>The required attribute specifies that an input field must be filled out before submitting the form.</p>`

- `<form action="/action_page.php">`
- `<label for="username">Username:</label>`
- `<input type="text" id="username" name="username" required>`
- `<input type="submit" value="Submit">`
- `</form>`

- `<p>Note: The required attribute of the input tag is not supported in Safari prior version 10.1.</p>`

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

<select>



- The <select> element is used to create a drop-down list.
- The <select> element is most often used in a form, to collect user input.
- The name attribute is needed to reference the form data after the form is submitted (if you omit the name attribute, no data from the drop-down list will be submitted).
- The id attribute is needed to associate the drop-down list with a label.



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The select element</h1>`

- `<p>The select element is used to create a drop-down list.</p>`

- `<form action="/action_page.php">`
- `<label for="cars">Choose a car:</label>`
- `<select name="cars" id="cars">`
- `<option value="volvo">Volvo</option>`
- `<option value="saab">Saab</option>`
- `<option value="opel">Opel</option>`
- `<option value="audi">Audi</option>`
- `</select>`
- `

`
- `<input type="submit" value="Submit">`
- `</form>`

- `<p>Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php".</p>`

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

<textarea>



- The <textarea> tag defines a multi-line text input control.
- The <textarea> element is often used in a form, to collect user inputs like comments or reviews.
- A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).
- The size of a text area is specified by the <cols> and <rows> attributes (or with CSS).



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The textarea element</h1>`

- `<form action="/action_page.php">`
- `<label >Review code:</label>`
- `<textarea id="review" name="w3review" rows="5" cols="100">`
- you will learn how to make a website. T
- `</textarea>`
- `

`
- `<input type="submit" value="Submit">`
- `</form>`

- `<p>Click the "Submit" button and the form-data will be sent to a page on the`
- `server .</p>`
- `</body>`
- `</html>`



- The `<output>` tag is used to represent the result of a calculation (like one performed by a script).



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The output element</h1>`

- `<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">`
- `<input type="NUMBER" id="a" value="50">`
- `+<input type="number" id="b" value="25">`
- `=<output name="x" for="a b"></output>`
- `</form>`

- `<p>Note: The output element is not supported in Edge 12 (or earlier).</p>`

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



- The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.
- The `<datalist>` tag is used to provide an "autocomplete" feature for `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.
- The `<datalist>` element's `id` attribute must be equal to the `<input>` element's `list` attribute (this binds them together).



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The datalist element</h1>`

- `<form action="/action_page.php" method="get">`
- `<label for="browser">Choose your browser from the list:</label>`
- `<input list="browsers" name="browser" id="browser">`
- `<datalist id="browsers">`
- `<option value="Edge">`
- `<option value="Firefox">`
- `<option value="Chrome">`
- `<option value="Opera">`
- `<option value="Safari">`
- `</datalist>`
- `<input type="submit">`
- `</form>`

- `<p>Note: The datalist tag is not supported in Safari 12.0 (or earlier).</p>`

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

<fieldset>



- The <fieldset> tag is used to group related elements in a form.
- The <fieldset> tag draws a box around the related elements.



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The fieldset element</h1>`

- `<form action="/action_page.php">`
- `<fieldset>`
- `<legend>Personalia:</legend>`
- `<label for="fname">First name:</label>`
- `<input type="text" id="fname" name="fname">

`
- `<label for="lname">Last name:</label>`
- `<input type="text" id="lname" name="lname">

`
- `<label for="email">Email:</label>`
- `<input type="email" id="email" name="email">

`
- `<label for="birthday">Birthday:</label>`
- `<input type="date" id="birthday" name="birthday">

`
- `<input type="submit" value="Submit">`
- `</fieldset>`
- `</form>`

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



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- `</head>`
- `<body>`

- `<h1>The picture element</h1>`

- `<p>Resize the browser window to load different images.</p>`

- `<picture>`
- `<source media="(min-width:650px)" srcset="img_pink_flowers.jpg">`
- `<source media="(min-width:465px)" srcset="img_white_flower.jpg">`
- ``
- `</picture>`

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

IMG Tag



- **Responsive Images**

- `<html>`
- `<head>`
- `<style>`
- `img {`
- `width: 100%;`
- `height: auto;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- ``
- `</body>`
- `</html>`

Ismap



- `<html>`
- `<head>`
- `<style>`
- `img {`
- `width: 100%;`
- `height: auto;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- ``
- ``
- `</body>`
- `</html>`



```
• <!doctype html>
• <html lang="en">

• <head>
• <meta charset="utf-8" />
• <title>Wolves</title>
• <link href='http://fonts.googleapis.com/css?family=UnifrakturCook:700' rel='stylesheet' type='text/css'>
• <style>
• * { box-sizing: border-box; }
• body, figure { margin: 0; }
• img { display: block; width: 100%; }

• body {
•     font-family: Avenir, Corbel, sans-serif;
•     text-align: center;
• }
• h1 {
•     font-family: 'UnifrakturCook', Avenir, Corbel, sans-serif;
•     margin: 0.125em;
•     font-size: 4em;
•     margin: 0.5em;
•     text-align: center;
• }

• figure {
•     padding: 1px;
• }

• figcaption {
•     display: block;
•     margin: 0;
•     text-align: center;
•     font-weight: bold;
•     color: white;
•     position: relative;
•     top: -3em;
•     text-shadow: 1px 1px 1px #000;
•     height: 0;
•     padding: 0;
•     font-size: 1.125em;
• }

• @media (min-width: 36em) {
•     figure {
•         display: block;
•         float: left;
•         width: 33.333333333333%;
•     }
• }
```

SVG



- SVG stands for Scalable Vector Graphics
 - SVG is used to define graphics for the Web
 - SVG is a W3C recommendation
-
- The `svg` element is a container that defines a new coordinate system and viewport. It is used as the outermost element of SVG documents, but it can also be used to embed a SVG fragment inside an SVG or HTML document.

Circle In SVG



- `<!DOCTYPE html>`
`<html>`
`<body>`

`<svg width="100" height="100">`
 `<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />`
`</svg>`

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

Rectangle In SVG



- `<!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>`

Rounded Rectangle



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<svg width="400" height="180">`
- `<rect x="50" y="20" rx="20" ry="20" width="150"`
`height="150"`
- `style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />`
- Sorry, your browser does not support inline SVG.
- `</svg>`

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

canvas



- The `<canvas>` tag is used to draw graphics, on the fly, via scripting (usually JavaScript).
- The `<canvas>` tag is transparent, and is only a container for graphics, you must use a script to actually draw the graphics.
- Any text inside the `<canvas>` element will be displayed in browsers with JavaScript disabled and in browsers that do not support `<canvas>`.



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The canvas element</h1>`

- `<canvas id="myCanvas">Your browser does not support the canvas tag.</canvas>`

- `<script>`
- `var c = document.getElementById("myCanvas");`
- `var ctx = c.getContext("2d");`
- `ctx.fillStyle = "#FF0000";`
- `ctx.fillRect(0, 0, 80, 100);`
- `</script>`

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



- `<!DOCTYPE html>`
- `<html>`
- `<body>`

- `<h1>The canvas element</h1>`

- `<canvas id="myCanvas">`
- Your browser does not support the canvas tag.
- `</canvas>`

- `<script>`
- `var c = document.getElementById("myCanvas");`
- `var ctx = c.getContext("2d");`
- `ctx.fillStyle = "red";`
- `ctx.fillRect(20, 20, 75, 50);`

- `//Turn transparency on`
- `ctx.globalAlpha = 0.2;`
- `ctx.fillStyle = "blue";`
- `ctx.fillRect(50, 50, 75, 50);`
- `ctx.fillStyle = "green";`
- `ctx.fillRect(80, 80, 75, 50);`
- `</script>`

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

Position in HTML



- The position Property
- The position property specifies the type of positioning method used for an element.
- There are five different position values:
 - static
 - relative
 - fixed
 - absolute
 - sticky
- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.



- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

Static



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `div.static {`
- `position: static;`
- `border: 3px solid #73AD21;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h2>position: static;</h2>`
- `<p>An element with position: static; is not positioned in any special way; it is`
- `always positioned according to the normal flow of the page:</p>`
- `<div class="static">`
- `This div element has position: static;`
- `</div>`
- `</body>`
- `</html>`

Fixed



- An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The `top`, `right`, `bottom`, and `left` properties are used to position the element.

Fixed



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `div.fixed {`
- `position: fixed;`
- `bottom: 0;`
- `right: 0;`
- `width: 300px;`
- `border: 3px solid #73AD21;`
- `}`
- `</style>`
- `</head>`
- `<body>`

- `<h2>position: fixed;</h2>`

- `<p>An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:</p>`

- `<div class="fixed">`
- This div element has position: fixed;
- `</div>`

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

Relative



- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `div.relative {`
- `position: relative;`
- `left: 30px;`
- `border: 3px solid #73AD21;`
- `}`
- `</style>`
- `</head>`
- `<body>`

- `<h2>position: relative;</h2>`

- `<p>An element with position: relative; is positioned relative to its normal position:</p>`

- `<div class="relative">`
- `This div element has position: relative;`
- `<h1>hello</h1>`
- `</div>`

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

Absolute



An element with `position: absolute`; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like `fixed`).

- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `div.relative {`
- `position: relative;`
- `width: 400px;`
- `height: 200px;`
- `border: 3px solid #73AD21;`
- `}`
- `div.absolute {`
- `position: absolute;`
- `top: 80px;`
- `right: 0;`
- `width: 200px;`
- `height: 100px;`
- `border: 3px solid #73AD21;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h2>position: absolute;</h2>`
- `<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed);</p>`
- `<div class="relative">This div element has position: relative;`
- `<div class="absolute">This div element has position: absolute;</div>`
- `</div>`
- `</body>`
- `</html>`

overflow



- The overflow property specifies what should happen if content overflows an element's box.
- This property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.
- **Note:** The overflow property only works for block elements with a specified height.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
  background-color: lightblue;
  width: 110px;
  height: 110px;
  overflow: scroll;
}
```



```
div.ex2 {
  background-color: lightblue;
  width: 110px;
  height: 110px;
  overflow: hidden;
}
```

```
div.ex3 {
  background-color: lightblue;
  width: 110px;
  height: 110px;
  overflow: auto;
}
```

```
div.ex4 {
  background-color: lightblue;
  width: 110px;
  height: 110px;
  overflow: visible;
}
```

```
</style>
</head>
<body>
```

```
<h1>The overflow Property</h1>
```

```
<p>The overflow property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.</p>
```

```
<h2>overflow: scroll:</h2>
```

```
<div class="ex1">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</div>
```

```
<h2>overflow: hidden:</h2>
```

```
<div class="ex2">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</div>
```

```
<h2>overflow: auto:</h2>
```

```
<div class="ex3">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</div>
```

```
<h2>overflow: visible (default):</h2>
```

```
<div class="ex4">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</div>
```

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



- The height property sets the height of an element.
- The height of an element does not include padding, borders, or margins!
- If height: auto; the element will automatically adjust its height to allow its content to be displayed correctly.

width



- The width property sets the width of an element.
- The width of an element does not include padding, borders, or margins!



```
<!DOCTYPE html>
<html>
<head>
<style>
div.a {
  width: auto;
  border: 1px solid black;
}

div.b {
  width: 150px;
  border: 1px solid black;
}

div.c {
  width: 50%;
  border: 1px solid black;
}
</style>
</head>
<body>
<h1>The width Property</h1>
```

```
<h2>width: auto (default)</h2>
<div class="a">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.
Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.</div>
```

```
<h2>width: 150px</h2>
<div class="b">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.
Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.</div>
```

```
<h2>width: 50%</h2>
<div class="c">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.
Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.</div>
```

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

Align



- The align-content property modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines.
- Note: There must be multiple lines of items for this property to have any effect!



```
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 70px;
  height: 300px;
  border: 1px solid #c3c3c3;
  display: flex;
  flex-wrap: wrap;
  align-content: center;
}
```

```
#main div {
  width: 70px;
  height: 70px;
}
</style>
</head>
<body>
```

```
<h1>The align-content Property</h1>
```

```
<div id="main">
  <div style="background-color:coral;"></div>
  <div style="background-color:lightblue;"></div>
  <div style="background-color:pink;"></div>
</div>
```

```
<p><b>Note:</b> Internet Explorer 10 and earlier versions do not support the align-content property.</p>
```

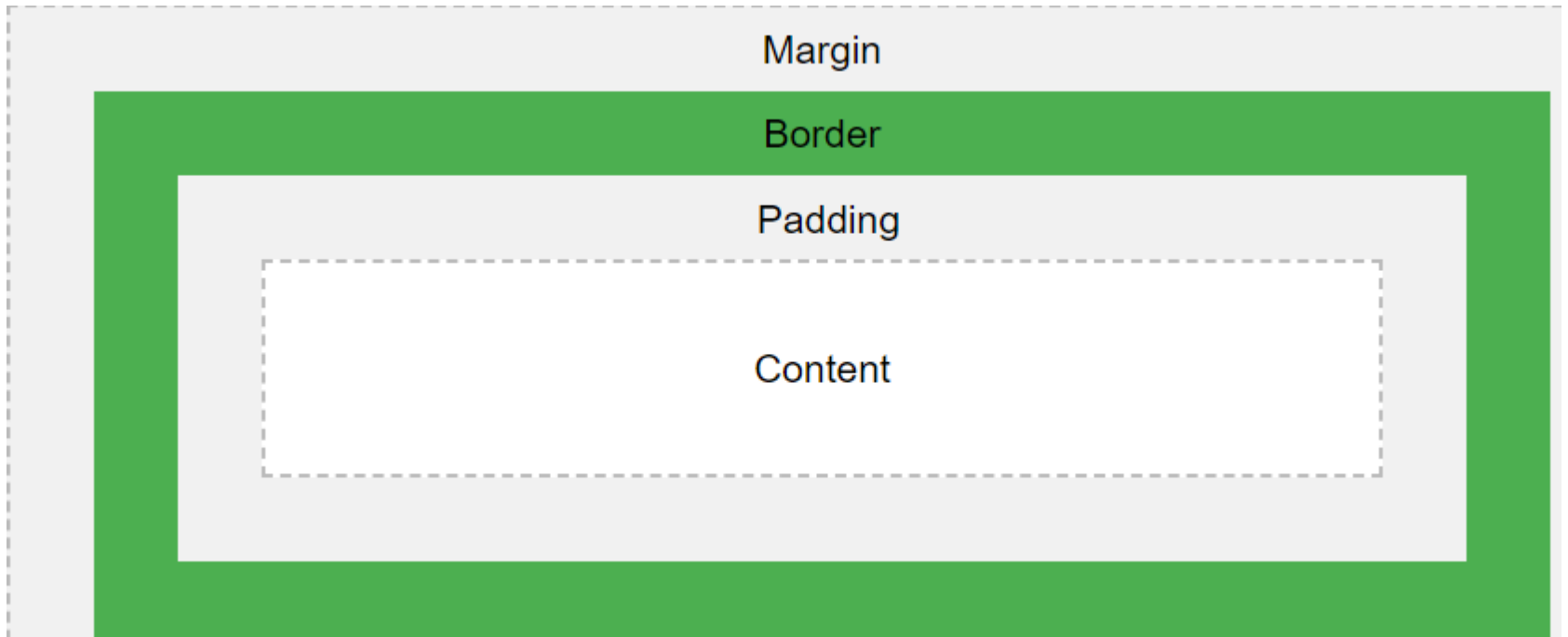
```
</body>
</html>
```




- **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





- 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

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`



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>
```

```
<h2>Using individual margin properties</h2>
```

```
<div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left
margin of 80px.</div>
```

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

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



- When **one** value is specified, it applies the same padding to **all four sides**.
- When **two** values are specified, the first padding applies to the **top and bottom**, the second to the **left and right**.
- When **three** values are specified, the first padding applies to the **top**, the second to the **right and left**, the third to the **bottom**.
- When **four** values are specified, the paddings apply to the **top, right, bottom, and left** in that order (clockwise).



- `<!doctype html>`
- `<html>`
- `<head>`
- `<title>Padding</title>`
- `<style>`
- `h4 { background-color: lime; padding: 20px 50px; } h3 { background-color: cyan; padding: 110px 50px 50px 110px; }`
- `</style>`
- `</head>`
- `<body>`
- `<h4>hello this is me</h4>`
- `<h3>Welcome to e- classes </h3>`
- `</body>`
- `</html>`

Font Family



- The font family of a text is set with the font-family property.
- The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.
- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.



- Sans-Serif
- Arial, sans-serif;
- Helvetica, sans-serif;
- Gill Sans, sans-serif;
- Lucida, sans-serif;
- Helvetica Narrow, sans-serif;
- sans-serif;
- Serif
- Times, serif;
- Times New Roman, serif;
- Palatino, serif;
- Bookman, serif;
- New Century Schoolbook, serif;
- serif;
- Monospace
- Andale Mono, monospace;
- Courier New, monospace;
- Courier, monospace;
- Lucidatypewriter, monospace;
- Fixed, monospace;
- monospace;
- Cursive
- Comic Sans, Comic Sans MS, cursive;
- Zapf Chancery, cursive;
- Coronetscript, cursive;
- Florence, cursive;
- Parkavenue, cursive;
- cursive;
- Fantasy
- Impact, fantasy;
- Arnoldboecklin, fantasy;
- Oldtown, fantasy;
- Blippo, fantasy;
- Brushstroke, fantasy;
- fantasy;



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `.serif {`
- `font-family: "Times New Roman", Times, serif;`
- `}`
- `.sansserif {`
- `font-family: Arial, Helvetica, sans-serif;`
- `}`
- `.monospace {`
- `font-family: "Lucida Console", Courier, monospace;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>CSS font-family</h1>`
- `<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>`
- `<p class="sansserif">This is a paragraph, shown in the Arial font.</p>`
- `<p class="monospace">This is a paragraph, shown in the Lucida Console font.</p>`
- `</body>`
- `</html>`

Font Size



- The font-size CSS property sets the size of the font. Changing the font size also updates the sizes of the font size-relative `<length>` units, such as em, ex, and so forth.
- There are several ways to specify the font size, including with keywords or numerical values for pixels or ems. Choose the appropriate method based on the needs for the particular web page.

Pixel



- Setting the font size in pixel values (px) is a good choice when you need pixel accuracy. A px value is static. This is an OS-independent and cross-browser way of literally telling the browsers to render the letters at exactly the number of pixels in height that you specified. The results may vary slightly across browsers, as they may use different algorithms to achieve a similar effect.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `h1 {`
- `font-size: 40px;`
- `}`
- `h2 {`
- `font-size: 30px;`
- `}`
- `p {`
- `font-size: 14px;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>This is heading 1</h1>`
- `<h2>This is heading 2</h2>`
- `<p>This is a paragraph.</p>`
- `<p>This is another paragraph.</p>`
- `</body>`
- `</html>`

EMS



- Another way of setting the font size is with em values. The size of an em value is dynamic. When defining the font-size property, an em is equal to the font size of the element on which the em is used. If you haven't set the font size anywhere on the page, then it is the browser default, which is often 16px. So, by default 1em = 16px, and 2em = 32px. If you set a font-size of 20px on the body element, then 1em = 20px and 2em = 40px. Note that the value 2 is essentially a multiplier of the current em size.
- In order to calculate the em equivalent for any pixel value required, you can use this formula:
 - $\text{em} = \text{desired element pixel value} / \text{parent element font-size in pixels}$
 - For example, suppose the font-size of the body of the page is set to 16px. If the font-size you want is 12px, then you should specify 0.75em (because $12/16 = 0.75$). Similarly, if you want a font size of 10px, then specify 0.625em ($10/16 = 0.625$); for 22px, specify 1.375em ($22/16$).



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `h1 {`
- `font-size: 2.5em; /* 40px/16=2.5em */`
- `}`
- `h2 {`
- `font-size: 1.875em; /* 30px/16=1.875em */`
- `}`
- `p {`
- `font-size: 0.875em; /* 14px/16=0.875em */`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>This is heading 1</h1>`
- `<h2>This is heading 2</h2>`
- `<p>This is a paragraph.</p>`
- `<p>Specifying the font-size in em allows all major browsers to resize the text.`
- `Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.</p>`
- `</body>`
- `</html>`

Rem



- rem values were invented in order to sidestep the compounding problem. rem values are relative to the root html element, not the parent element. In other words, it lets you specify a font size in a relative fashion without being affected by the size of the parent, thereby eliminating compounding.
- The CSS below is nearly identical to the previous example. The only exception is that the unit has been changed to rem.
- ```
html {
 font-size: 62.5%; /* font-size 1em = 10px on default browser settings */
}
span {
 font-size: 1.6rem;
}
```



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `h1 {`
- `font-size: 40rem;`
- `}`
- `h2 {`
- `font-size: 30rem;`
- `}`
- `p {`
- `font-size: 14rem;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>This is heading 1</h1>`
- `<h2>This is heading 2</h2>`
- `<p>This is a paragraph.</p>`
- `<p>This is another paragraph.</p>`
- `</body>`
- `</html>`

# Font Weight



- Control the font weight of an element using the .
- font-{
- Weight
- } utilities.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p.normal {`
- `font-weight: normal;`
- `}`
- `p.light {`
- `font-weight: lighter;`
- `}`
- `p.thick {`
- `font-weight: bold;`
- `}`
- `p.thicker {`
- `font-weight: 900;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>The font-weight Property</h1>`
- `<p class="normal">This is a paragraph.</p>`
- `<p class="light">This is a paragraph.</p>`
- `<p class="thick">This is a paragraph.</p>`
- `<p class="thicker">This is a paragraph.</p>`
- `</body>`
- `</html>`

# Link Style



- 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



- <!DOCTYPE html>
- <html>
- <head>
- <style>
- /\* 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;
- }
- </style>
- </head>
- <body>
- 
- <p><b><a href="https://www.google.com" target="\_blank">This is a link</a></b></p>
- <p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
- <p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>
- 
- </body>
- </html>



# Text-decoration



- The text-decoration property is used to add "decorations" to inline content.
- Possible Values
- none – No decoration should be added to the inline text.
- underline – An underline is drawn beneath the inline text.
- overline – An overline is drawn above the inline text.
- line-through – A line should be drawn through the middle of the inline text.
- blink – The inline text should blink on and off, analogous to the BLINK element introduced by Netscape supported by Netscape Version 5.0 web browser.



- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `h1 {`
- `text-decoration: blink;`
- `}`
- `h2 {`
- `text-decoration: line-through;`
- `}`
- `h3 {`
- `text-decoration: underline;`
- `}`
- `h4 {`
- `text-decoration: underline overline;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1>This is heading 1</h1>`
- `<h2>This is heading 2</h2>`
- `<h3>This is heading 3</h3>`
- `<h4>This is heading 4</h4>`
- `</body>`
- `</html>`

# Text-indent



- The text-indent property in CSS is used to define the indentation of the first line in each block of text. It also take negative values. It means if the value is negative then the first line will be indented to the left.



```
• <!DOCTYPE html>
• <html>
• <head>
• <title>
• CSS text-indent Property
• </title>

• <!-- CSS text-indent property -->
• <style>
• .s {
• text-indent: 70px; }

• .g {
• text-indent: 1em;
• }

• .h {
• text-indent: 40%;
• }
• </style>
• </head>
• <body>
• <h1 style = "">GeeksforGeeks</h1>
• <h2> text-indent Property</h2>

• <h2>text-indent: 70px:</h2>
• <div class = "s">
• Prepare for the Recruitment drive of product
• based companies like Microsoft, Amazon, Adobe
• etc with a free online placement preparation
• course.
• </div>

• <h2>text-indent: -5em:</h2>
• <div class = "g">
• Prepare for the Recruitment drive of product
• based companies like Microsoft, Amazon, Adobe
• etc with a free online placement preparation
• course.
• </div>

• <h2>text-indent: 40%:</h2>
• <div class = "h">
• Prepare for the Recruitment drive of product
• based companies like Microsoft, Amazon, Adobe
• etc with a free online placement preparation
• course.
• </div>
• </body>
• </html>
```

# Border Color



- color Specifies the color of the border. It must be a valid color. You can provide between one and four color values. If you provide all four values, these set the top, right, bottom and left border, respectively.

# One Colour Definition



- `<!doctype html>`
- `<title>Example</title>`
- `<style>`
- `div {`
- `font-size: 1.2em;`
- `padding: 20px;`
- `border-width: 9px;`
- `border-style: solid;`
- `border-color: cyan;`
- `}`
- `</style>`
- `<div>`
- `<code>border-color: cyan;</code>`
- `</div>`

# Two Colour Definition



- `<!doctype html>`
- `<title>Example</title>`
- `<style>`
- `div {`
- `font-size: 1.2em;`
- `padding: 20px;`
- `border-width: 9px;`
- `border-style: solid;`
- `border-color: cyan magenta;`
- `}`
- `</style>`
- `<div>`
- `<code>border-color: cyan magenta;</code>`
- `</div>`

# Three Colour Definition



- `<!doctype html>`
- `<title>Example</title>`
- `<style>`
- `div {`
- `font-size: 1.2em;`
- `padding: 20px;`
- `border-width: 9px;`
- `border-style: solid;`
- `border-color: cyan magenta yellow;`
- `}`
- `</style>`
- `<div>`
- `<code>border-color: cyan magenta yellow;</code>`
- `</div>`



# Four Colour Definition



- `<!doctype html>`
- `<title>Example</title>`
- `<style>`
- `div {`
- `font-size: 1.2em;`
- `padding: 20px;`
- `border-width: 9px;`
- `border-style: solid;`
- `border-color: cyan magenta yellow black;`
- `}`
- `</style>`
- `<div>`
- `<code>border-color: cyan magenta yellow black;</code>`
- `</div>`

# Border-width



- The border-width property may be specified using one, two, three, or four values.
- When **one** value is specified, it applies the same width to **all four sides**.
- When **two** values are specified, the first width applies to the **top and bottom**, the second to the **left and right**.
- When **three** values are specified, the first width applies to the **top**, the second to the **left and right**, the third to the **bottom**.
- When **four** values are specified, the widths apply to the **top, right, bottom, and left** in that order (clockwise).



```
• <!doctype html>
• <title>Example</title>
• <style>
• #sval {
• border: ridge #ccc;
• border-width: 6px;
• }
• #bival {
• border: solid red;
• border-width: 2px 10px;
• }
• #treval {
• border: dotted orange;
• border-width: 0.3em 0 9px;
• }
• #fourval {
• border: solid lightgreen;
• border-width: thin medium thick 1em;
• }
• p {
• width: auto;
• margin: 0.25em;
• padding: 0.25em;
• }
• </style>
• <body>
• <p id="sval">
• one value: 6px wide border on all 4 sides</p>
• <p id="bival">
• two different values: 2px wide top and bottom border, 10px wide right and left border</p>
• <p id="treval">
• three different values: 0.3em top, 9px bottom, and zero width right and left</p>
• <p id="fourval">
• four different values: "thin" top, "medium" right, "thick" bottom, and 1em left</p>
• </body>
• </html>
```

# Background



- The background property in CSS allows you to control the background of any element (what paints underneath the content in that element). It is a shorthand property, which means that it allows you to write what would be multiple CSS properties in one



- background is made up of eight properties:
- background-image
- background-position
- background-size
- background-repeat
- background-attachment
- background-origin
- background-clip
- background-color

# Background-image



- The background-image property in CSS applies a graphic (e.g. PNG, SVG, JPG, GIF, WEBP) or gradient to the background of an element.
- There are two different types of images you can include with CSS: regular images and gradients.



- `body {`
- `background: url(IR.jpg);`
- `}`

# Gradients



- Another option when using backgrounds is to tell the browser to create a gradient
- `body { background: linear-gradient(black, white); }`
- `body { background: radial-gradient(circle, black, white); }`



# Setting a Fallback Color



- If a background image fails to load, or your gradient background is viewed on a browser that doesn't support gradients, the browser will look for a background color as a fallback.
- `body { background: url(sweettexture.jpg) blue; }`

# Background Clip



- `<!doctype html>`
- `<title>Example</title>`
- `<style>`
- `@import url(https://fonts.googleapis.com/css?family=Syncopate);`
- `body {`
- `background: black;`
- `text-align: center;`
- `padding: 120px 20px;`
- `}`
- `h1 {`
- `display: inline-block;`
- `font-size: 80px;`
- `line-height: 0.9;`
- `padding: 20px;`
- `font-family: 'Syncopate', sans-serif;`
- `text-transform: uppercase;`
- `background: radial-gradient(`
- `circle farthest-corner at center center,`
- `white,`
- `#111`
- `)`
- `}`
- `</style>`
- `</body>`
- `<H1>HELLO</H1> </body>`
- `</html>`



# UNIT OF MEASUREMENT



- % Defines a measurement as a percentage relative to another value, typically an enclosing element. `p {font-size: 16pt; line-height: 125%;}`
- cm Defines a measurement in centimeters. `div {margin-bottom: 2cm;}`
- em A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt. `p {letter-spacing: 7em;}`
- ex This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x. `p {font-size: 24pt; line-height: 3ex;}`
- in Defines a measurement in inches. `p {word-spacing: .15in;}`
- mm Defines a measurement in millimeters. `p {word-spacing: 15mm;}`
- pc Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch. `p {font-size: 20pc;}`
- pt Defines a measurement in points. A point is defined as 1/72nd of an inch. `body {font-size: 18pt;}`
- px Defines a measurement in screen pixels. `p {padding: 25px;}`



- Responsive effects with css



- Responsive web design makes your web page look good on all devices.
- Responsive web design uses only HTML and CSS.
- Responsive web design is not a program or a JavaScript.

# Boot Strap









- <!DOCTYPE html>
- <html>
- <head>
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- <style>
- \* {
- box-sizing: border-box;
- }

- .row::after {
- content: "";
- clear: both;
- display: table;
- }

- [class\*="col-"] {
- float: left;
- padding: 15px;
- }

- html {
- font-family: "Lucida Sans", sans-serif;
- }

- .header {
- background-color: #9933cc;
- color: #ffffff;
- padding: 15px;
- }

- .menu ul {
- list-style-type: none;
- margin: 0;
- padding: 0;
- }

- .menu li {
- padding: 8px;
- margin-bottom: 7px;
- background-color: #33b5e5;
- color: #ffffff;
- box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);
- }

- .menu li:hover {
- background-color: #0099cc;
- }

- .aside {
- background-color: #33b5e5;



- **Templates Work**

# Links



- <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/>
- <https://www.htmldog.com/guides/css/beginner>
- <https://www.tutorialspoint.com/html/index.htm>

# End



Thank you for joining us.

Regards:

- Sushant Srivasatava
- (Sr. Software Developer)
- 7234999940
- [sushantdeveloper@rediffmail.com](mailto:sushantdeveloper@rediffmail.com)
- [sushantdeveloper@gmail.com](mailto:sushantdeveloper@gmail.com)