Introduction PA

Home Pratiksha

about us Pritam

product Pradnya

services Pritam

contact Pratiksha

our team Pradnya

vote of thanks Pritam

Section tag -The <section> tag defines a section in a document.

ID class- Difference between id and class attribute: The only difference between them is that**“id” is unique in a page and can only apply to at most one element**, while “class” selector can apply to multiple elements.

Div Span-Both the tags ([<div>](https://www.geeksforgeeks.org/div-tag-html/) and [<span>](https://www.geeksforgeeks.org/span-tag-html/)) are used to represent the part of the webpgage, <div> tag is used a as block part of the webpage and <span> tag is used as a inline part of the webpage

Div Tag-**HTML <div> tag:** The div tag is known as Division tag. The div tag is used in HTML to make divisions of content on the web page like (text, images, header, footer, navigation bar, etc). Div tag has both opening(<div>) and closing (</div>) tags and it is mandatory to close the tag. As we know Div tag is a block-level tag. In this example, the div tag contains the entire width. It will be displayed div tag each time on a new line, not on the same line.

**What Exactly Is CSS?**

Cascading Style Sheets (CSS) is a markup language responsible for how your web pages will look like. It controls the colors, fonts, and layouts of your website elements.

This style sheet language also allows you to add effects or animations to your website. You can use it to display some CSS animations like click button effects, spinners or loaders, and animated backgrounds.

Without CSS, your website will appear as a plain [**HTML page**](https://www.hostinger.com/tutorials/what-is-html). Here’s how [**Twitter**](https://twitter.com/) will look like if we disable its CSS:

There are three ways you can use to [**implement CSS into your HTML**](https://www.hostinger.com/tutorials/website/how-to-link-a-stylesheet-css-file-to-your-html-file): internal, external, and inline styles.

### Internal CSS

Internal or embedded CSS requires you to add **<style>** tag in the **<head>** section of your HTML document.

This CSS style is an effective method of styling a single page. However, using this style for multiple pages is time-consuming as you need to put CSS rules on every page of your website.

Here’s how you can use internal CSS:

1. Open your HTML page and locate **<head>** opening tag.
2. Put the following code right after the **<head>** tag

<style type="text/css">

1. Add CSS rules on a new line. Here’s an example:

body {

background-color: blue;

}

h1 {

color: red;

padding: 60px;

}

1. Type the closing tag:

</style>

Your HTML file will look like this:

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-color: blue;

}

h1 {

color: red;

padding: 60px;

}

</style>

</head>

<body>

<h1>Hostinger Tutorials</h1>

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

</body>

</html>

### External CSS

With external CSS, you’ll link your web pages to an external **.css** file, which can be created by any text editor in your device (e.g., [**Notepad++**](https://notepad-plus-plus.org/)).

This CSS type is a more efficient method, especially for styling a large website. By editing one **.css** file, you can change your entire site at once.

Follow these steps to use external CSS:

1. Create a new **.css** file with the text editor, and add the style rules. For example:

.xleftcol {

float: left;

width: 33%;

background:#809900;

}

.xmiddlecol {

float: left;

width: 34%;

background:#eff2df;

}

1. In the **<head>** section of your HTML sheet, add a reference to your external **.css** file right after **<title>** tag:

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

Don’t forget to change **style.css** with the name of your **.css**file.

### Inline CSS

Inline CSS is used to style a specific HTML element. For this CSS style, you’ll only need to add the **style** attribute to each HTML tag, without using selectors.

This CSS type is not really recommended, as each HTML tag needs to be styled individually. Managing your website may become too hard if you only use inline CSS.

However, inline CSS in HTML can be useful in some situations. For example, in cases where you don’t have access to CSS files or need to apply styles for a single element only.

Let’s take a look at an example. Here, we add an inline CSS to the **<p>** and **<h1>** tag:

<!DOCTYPE html>

<html>

<body style="background-color:black;">

<h1 style="color:white;padding:30px;">Hostinger Tutorials</h1>

<p style="color:white;">Something usefull here.</p>

</body>

</html>

* **Internal or embedded** ⁠— add **<style>** tag in the **<head>** section of HTML document
* **External** ⁠— link the HTML sheet to a separate **.css** file
* **Inline** ⁠— apply CSS rules for specific elements.

Img-The <img> tag is used to embed an image in an HTML page.

The <img> tag has two required attributes:

src - Specifies the path to the image

alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed.

1em = 16px (1 \* 16) scalable em

1rem = 16px scalable rem

100% = 16px Not scalable px

The flex property sets the flexible length on flexible items.

flex:1 1 20rem;

|  |  |
| --- | --- |
| [*flex-grow*](https://www.w3schools.com/cssref/css3_pr_flex-grow.asp) | A number specifying how much the item will grow relative to the rest of the flexible items |
| [*flex-shrink*](https://www.w3schools.com/cssref/css3_pr_flex-shrink.asp) | A number specifying how much the item will shrink relative to the rest of the flexible items |
| [*flex-basis*](https://www.w3schools.com/cssref/css3_pr_flex-basis.asp) | The length of the item. Legal values: "auto", "inherit", or a number followed by "%", "px", "em" or any other length unit |

box-shadow:0 .3rem .6rem #8fce00;

The box-shadow property in CSS is for putting shadows on elements

syntax is:

box-shadow: [horizontal offset] [vertical offset] [blur radius] [optional spread radius] [color];

1. **The horizontal offset** (required) of the shadow, positive means the shadow will be on the right of the box, a negative offset will put the shadow on the left of the box.
2. **The vertical offset** (required) of the shadow, a negative one means the box-shadow will be above the box, a positive one means the shadow will be below the box.
3. **The blur radius** (required), if set to 0 the shadow will be sharp, the higher the number, the more blurred it will be, and the further out the shadow will extend. For instance a shadow with 5px of horizontal offset that also has a 5px blur radius will be 10px of total shadow.
4. **Color** (required) – takes any color value, like hex, named, [rgba](https://css-tricks.com/rgba-browser-support/) or [hsla](https://css-tricks.com/yay-for-hsla/). If the color value is omitted, box shadows are drawn in the foreground color (text color). But be aware, older WebKit browsers (pre Chrome 20 and Safari 6) ignore the rule when color is omitted.

Hover

The :hover selector is used to select elements when you mouse over them.

The translateY () CSS function **repositions an element vertically on the 2D plane**.

Header Tag

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information

head is the tag at the top of your page containing your meta-tags, styles, scripts and title. Header : header tag contains information related to the title and heading related content which is display on web page.

The <nav> tag defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

|  |  |  |
| --- | --- | --- |
| **Class** | | **Description** |
| .carousel | | Creates a carousel |
| .carousel-indicators | | Adds indicators for the carousel. These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing) |
| .carousel-inner | | Adds slides to the carousel |
| .carousel-item | | Specifies the content of each slide |
| .carousel-control-prev | | Adds a left (previous) button to the carousel, which allows the user to go back between the slides |
| .carousel-control-next | | Adds a right (next) button to the carousel, which allows the user to go forward between the slides |
| .carousel-control-prev-icon | Used together with .carousel-control-prev to create a "previous" button |
| .carousel-control-next-icon | | Used together with .carousel-control-next to create a "next" button |
| .slide | | Adds a CSS transition and animation effect when sliding from one item to the next. Remove this class if you do not want this effect |

Ride

Autoplays the carousel after the user manually cycles the first item. If "carousel", autoplays the carousel on load.

Use data attributes to easily control the position of the carousel. data-bs-slide accepts the keywords prev or next, which alters the slide position relative to its current position. Alternatively, use data-bs-slide-to to pass a raw slide index to the carousel data-bs-slide-to="2", which shifts the slide position to a particular index beginning with 0.

The data-bs-ride="carousel" attribute is used to mark a carousel as animating starting at page load. If you don’t use data-bs-ride="carousel" to initialize your carousel, you have to initialize it yourself. It cannot be used in combination with (redundant and unnecessary) explicit JavaScript initialization of the same carousel.

Here are the major differences between GET and POST:

|  |  |
| --- | --- |
| **GET** | **POST** |
| In GET method, values are visible in the URL. | In POST method, values are not visible in the URL. |
| GET has a limitation on the length of the values, generally 255 characters. | POST has no limitation on the length of the values since they are submitted via the body of HTTP. |
| GET performs are better compared to POST because of the simple nature of appending the values in the URL. | It has lower performance as compared to GET method because of time spent in including POST values in the HTTP body. |
| This method supports only string data types. | This method supports different data types, such as string, numeric, binary, etc. |
| GET results can be bookmarked. | POST results cannot be bookmarked. |
| GET request is often cacheable. | The POST request is hardly cacheable. |
| GET Parameters remain in web browser history. | Parameters are not saved in web browser history. |

**Advantages of POST**

Here, are benefits/ pros of using POST:

* This method helps you to determine resource URI.
* Specifying a new resource location header is very easy using location header.
* You can send a request to accept the entity as a new resource, which is identified by the URI.
* You can send user-generated data to the web server.
* It is very useful when you do not have any idea about the resource you have to keep in the URL.
* Use POST when you need the server, which controls URL generation of your resources.
* POST is a secure method as its requests do not remain in browser history.
* You can effortlessly transmit a large amount of data using post.
* You can keep the data private.
* This method can be used to send binary as well as ASCII data.

The function bigImg() is triggered when the user mouse over the image. This function enlarges the image.

The function normalImg() is triggered when the mouse pointer is moved out of the image. That function sets the height and width of the image back to normal.

## Definition and Usage

## :root{

## --main-color:#fa4a49;

## }

The :root selector matches the document's root element.

In HTML, the root element is always the html element.

\*- Selects all elements

**transition: background.2s linear;** This says, transition the background defined on is element’s original state to the background defined on this state over.2 seconds in a linear fashion. The order is important as the first time value id assigned to the transition-duration, and the second time value is assigned to transition-delay.

|  |  |
| --- | --- |
| vh | Relative to 1% of the height of the viewport\* |