CSS Tutorial

CSS tutorial or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

CSS Example with CSS Editor

In this tutorial, you will get a lot of CSS examples, you can edit and run these examples with our online CSS editor tool.

```
<!DOCTYPE>
<html>
<head>
<style>
h1{
color:white;
background-color:red;
padding:5px;
}
p{
color:blue;
}
</style>
</head>
<body>
<h1>Write Your First CSS Example</h1>
<p>This is Paragraph.</p>
</body>
</html>
```

This is Paragraph.

CSS Index

CSS Tutorial

CSS Properties

CSS Advance

CSS Design

Interview Questions

CSS 3 Tutorial

In this tutorial, we will learn CSS 3 properties to design box model, apply opacity, radius etc.

All CSS Properties

In this tutorial, you will get details of all CSS properties such as background, border, font, float, display, margin, opacity, padding, text-align, vertical-align, position, color etc.

Prerequisite

Before learning CSS, you must have the basic knowledge of HTML.

Audience

Our CSS tutorial is designed to help beginners and professionals both.

Problem

If you find any problem or mistake in our CSS tutorial, you can report to us. We assure, you will not find any problem in CSS tutorial.

What is CSS

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

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

What does CSS do

- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS code.

Why use CSS

These are the three major benefits of CSS:

1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

2) Saves a lot of time

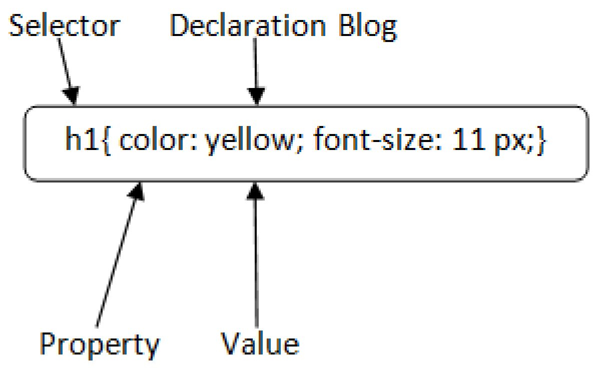
CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

CSS Syntax

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



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

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

1. color: yellow;
2. font-size: 11 px;

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

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

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

1. Selector{Property1: value1; Property2: value2; ....;}

CSS Selector

CSS selectors are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
   text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

Test it Now

Output:

This style will be applied on every paragraph.

Me too!

And me!

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let?s take an example with the id "para1".

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Javatpoint.com</p>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

Test it Now

Output:

Hello Javatpoint.com

This paragraph will not be affected.

3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

Note: A class name should not be started with a number.

Let's take an example with a class "center".

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
```

```
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

Test it Now

Output:

This heading is blue and center-aligned.

This paragraph is blue and center-aligned.

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

Test it Now

Output:

This heading is not affected

This paragraph is blue and center-aligned.

4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
   color: green;
   font-size: 20px;
}
```

```
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

Test it Now

Output:

This is heading

This style will be applied on every paragraph.

Me too!

And me!

5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

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

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

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

Let's see the full example of CSS group selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
<h2>Hello Javatpoint.com (In smaller font)</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

Test it Now

Output:

Hello Javatpoint.com

Hello Javatpoint.com (In smaller font)

This is a paragraph.

How to add CSS

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

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

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

```
<p style="color:blue">Hello CSS</p>
```

2) Internal CSS

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

For example:

```
<style>
p{color:blue}
</style>
```

3) External CSS

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

For example:

```
p{color:blue}
```

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

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

The link tag must be used inside head section of html.

Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

```
<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>
```

Example:

```
<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h2>
<p>This paragraph is not affected.</p>
```

Output:

Inline CSS is applied on this heading.

This paragraph is not affected.

Disadvantages of Inline CSS

- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

Internal CSS

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: red;
    margin-left: 80px;
}
</style>
</head>
<body>
<h1>The internal style sheet is applied on this heading.</h1>
```

```
<p>This paragraph will not be affected.</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.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

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

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML element. Let's take an example of a style sheet file named "mystyle.css".

File: mystyle.css

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

Note: You should not use a space between the property value and the unit. For example: It should be margin-left:20px not margin-left:20 px.

CSS Comments

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.

Comments are ignored by browsers.

Comments are single or multiple lines statement and written within /*............*/ .

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: blue;
    /* This is a single-line comment */
    text-align: center;
}
/* This is
a multi-line
comment */
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>This statement is styled with CSS.</p>
```

```
<p>CSS comments are ignored by the browsers and not shown in the output.</p>
</body>
</html>
```

Output:

Hello Javatpoint.com

This statement is styled with CSS.

CSS comments are ignored by the browsers and not shown in the output.

CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

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

1) CSS background-color

The background-color property is used to specify the background color of the element.

You can set the background color like this.,

```
<!DOCTYPE html>
<html>
<head>
<style>
h2,p{
    background-color: #b0d4de;
}
</style>
</head>
<body>
<h2>My first CSS page.</h2>
<p>Hello Javatpoint. This is an example of CSS background-color.</p>
</body>
</html>
```

Output:

My first CSS page.

Hello Javatpoint. This is an example of CSS background-color.

2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
body {
background-image: url("paper1.gif");
margin-left:100px;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
</body>
</html>
```

Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

3) CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

The background looks better if the image repeated horizontally only.

background-repeat: repeat-x;

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
</body>
</html>
```

background-repeat: repeat-y;

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-y;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
</body>
</html>
```

4) CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let?s take an example with fixed background image.

1. background: white url('bbb.gif');
2. background-repeat: no-repeat;
3. background-attachment: fixed;

5) CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

1. center
2. top
3. bottom
4. left
5. right

1. background: white url('good-morning.jpg');
2. background-repeat: no-repeat;
3. background-attachment: fixed;
4. background-position: center;

CSS Border

The CSS border is a shorthand property used to set the border on an element.

The CSS border properties are use to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color
- border-width
- border-radius

1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.

double	It defines two borders wlth the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.none {border-style: none;}
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.hidden {border-style: hidden;}
</style>
</head>
<body>
<p class="none">No border.</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="hidden">A hidden border.</p>
</body>
</html>
```

Output:

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

> ***Note: The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.***

```
<!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: solid;
    border-width: 1px;
}
</style>
</head>
<body>
<p class="one">Write your text here.</p>
<p class="two">Write your text here.</p>
<p class="three">Write your text here.</p>
</body>
</html>
```

3) CSS border-color

There are three methods to set the color of the border.

- Name: It specifies the color name. For example: "red".
- RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

> ***Note: The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.***

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-color: red;
}
p.two {
    border-style: solid;
    border-color: #98bf21;
}
</style>
</head>
<body>
<p class="one">This is a solid red border</p>
<p class="two">This is a solid green border</p>
</body>
</html>
```

CSS Display

CSS display is the most important property of CSS which is used to control the layout of the element. It specifies how the element is displayed.

Every element has a default display value according to its nature. Every element on the webpage is a rectangular box and the CSS property defines the behavior of that rectangular box.

CSS Display default properties

default value	inline
inherited	no
animation supporting	no
version	css1
javascript syntax	object.style.display="none"

Syntax

```
display:value;
```

CSS display values

There are following CSS display values which are commonly used.

1. display: inline;
2. display: inline-block;
3. display: block;
4. display: run-in;
5. display: none;

1) CSS display inline

The inline element takes the required width only. It doesn't force the line break so the flow of text doesn't break in inline example.

The inline elements are:

- <span>
- <a>
- <em>
- <b> etc.

Let's see an example of CSS display inline.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
display: inline;
}
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>Java Tutorial.</p>
<p>SQL Tutorial.</p>
<p>HTML Tutorial.</p>
<p>CSS Tutorial.</p>
</body>
</html>
```

Output:

Hello Javatpoint.com Java Tutorial. SQL Tutorial. HTML Tutorial. CSS Tutorial.

2) CSS display inline-block

The CSS display inline-block element is very similar to inline element but the difference is that you are able to set the width and height.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
display: inline-block;
}
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>Java Tutorial.</p>
<p>SQL Tutorial.</p>
<p>HTML Tutorial.</p>
<p>CSS Tutorial.</p>
</body>
</html>
```

Output:

Hello Javatpoint.com Java Tutorial. SQL Tutorial. HTML Tutorial. CSS Tutorial.

3) CSS display block

The CSS display block element takes as much as horizontal space as they can. Means the block element takes the full available width. They make a line break before and after them.

```
<!DOCTYPE html>
<html>
```

```
<head>
<style>
p {
display: block;
}
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>Java Tutorial.</p>
<p>SQL Tutorial.</p>
<p>HTML Tutorial.</p>
<p>CSS Tutorial.</p>
</body>
</html>
```

Output:

Hello Javatpoint.com

Java Tutorial.

SQL Tutorial.

HTML Tutorial.

CSS Tutorial.

4) CSS display run-in

This property doesn't work in Mozilla Firefox. These elements don't produce a specific box by themselves.

- If the run-in box contains a bock box, it will be same as block.
- If the block box follows the run-in box, the run-in box becomes the first inline box of the block box.
- If the inline box follows the run-in box, the run-in box becomes a block box.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
display: run-in;
}
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>Java Tutorial.</p>
<p>SQL Tutorial.</p>
<p>HTML Tutorial.</p>
<p>CSS Tutorial.</p>
</body>
</html>
```

Output:

Hello Javatpoint.com

Java Tutorial.

SQL Tutorial.

HTML Tutorial.

CSS Tutorial.

5) CSS display none

The "none" value totally removes the element from the page. It will not take any space.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
    display: none;
}
</style>
</head>
<body>
<h1>This heading is visible.</h1>
<h1 class="hidden">This is not visible.</h1>
<p>You can see that the hidden heading does not contain any space.</p>
</body>
</html>
```

Output:

```
This heading is visible.

You can see that the hidden heading does not contain any space.
```

Other CSS display values

Property-value	Description
flex	It is used to display an element as an block-level flex container. It is new in css3.
inline-flex	It is used to display an element as an inline-level flex container. It is new in css3.
inline-table	It displays an element as an inline-level table.
list-Item	It makes the element behave like a <li> element.
table	It makes the element behave like a <table> element.
table-caption	It makes the element behave like a <caption> element.
table-column-group	It makes the element behave like a <colgroup> element.
table-header-group	It makes the element behave like a <thead> element.
table-footer-group	It makes the element behave like a <tfoot> element.

table-row-group	It makes the element behave like a <tbody> element.
table-cell	It makes the element behave like a <td> element.
table-row	It makes the element behave like a <tr> element.
table-column	It makes the element behave like a <col> element.

CSS Font

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more. You have already studied how to make text bold or underlined. Here, you will also know how to resize your font using percentage.

These are some important font attributes:

1. **CSS Font color**: This property is used to change the color of the text. (standalone attribute)
2. **CSS Font family**: This property is used to change the face of the font.
3. **CSS Font size**: This property is used to increase or decrease the size of the font.
4. **CSS Font style**: This property is used to make the font bold, italic or oblique.
5. **CSS Font variant**: This property creates a small-caps effect.
6. **CSS Font weight**: This property is used to increase or decrease the boldness and lightness of the font.

---

1) CSS Font Color

CSS font color is a standalone attribute in CSS although it seems that it is a part of CSS fonts. It is used to change the color of the text.

CSS Colors

The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.

In CSS, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.

We can define the color of an element by using the following ways:

- RGB format.
- RGBA format.
- Hexadecimal notation.
- HSL.
- HSLA.
- Built-in color.

Let's understand the syntax and description of the above ways in detail.

RGB Format

RGB format is the short form of '**RED GREEN** and **BLUE**' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.

The color values in this format are specified by using the **rgb()** property. This property allows three values that can either be in percentage or integer (range from 0 to 255).

This property is not supported in all browsers; that's why it is not recommended to use it.

Syntax

1. color: rgb(R, G, B);

RGBA Format

It is almost similar to RGB format except that **RGBA** contains **A (Alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** is for fully transparent, and **1.0** is for not transparent.

Syntax

1. color:rgba(R, G, B, A);

Hexadecimal notation

Hexadecimal can be defined as a six-digit color representation. This notation starts with the **# symbol** followed by six characters ranges from **0 to F**. In hexadecimal notation, the first two digits represent the **red (RR)** color value, the next two digits represent the **green (GG)** color value, and the last two digits represent the **blue (BB)** color value.

The black color notation in hexadecimal is #000000, and the white color notation in hexadecimal is #FFFFFF. Some of the codes in hexadecimal notation are #FF0000, #00FF00, #0000FF, #FFFF00, and many more.

Syntax

1. color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

Short Hex codes

It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.

For example, #7B6 becomes #77BB66 in hexadecimal.

The black color notation in short hex is #000, and the white color notation in short hex is #FFF. Some of the codes in short hex are #F00, #0F0, #0FF, #FF0, and many more.

HSL

It is a short form of **Hue, Saturation,** and **Lightness**. Let's understand them individually.

Hue: It can be defined as the degree on the color wheel from 0 to 360. 0 represents red, 120 represents green, 240 represents blue.

Saturation: It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.

Lightness: The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

Let's see the syntax of HSL in color property.

Syntax

1. color:hsl(H, S, L);

HSLA

It is entirely similar to HSL property, except that it contains **A (alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** indicates fully transparent, and **1.0** indicates not transparent.

Syntax

1. color:hsla(H, S, L, A);

Built-in Color

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

Syntax

```
color: color-name;
```

Let's see the list of built-in colors along with their decimal and hexadecimal values.

S.no.	Color name	Hexadecimal Value	Decimal Value or rgb() value
1.	Red	#FF0000	rgb(255,0,0)
2.	Orange	#FFA500	rgb(255,165,0)
3.	Yellow	#FFFF00	rgb(255,255,0)
4.	Pink	#FFC0CB	rgb(255,192,203)
5.	Green	#008000	rgb(0,128,0)
6.	Violet	#EE82EE	rgb(238,130,238)
7.	Blue	#0000FF	rgb(0,0,255)
8.	Aqua	#00FFFF	rgb(0,255,255)
9.	Brown	#A52A2A	rgb(165,42,42)
10.	White	#FFFFFF	rgb(255,255,255)
11.	Gray	#808080	rgb(128,128,128)
12.	Black	#000000	rgb(0,0,0)

The illustration of CSS colors, which includes the above properties, is given below.

Example

```
<html>
   <head>
       <title>CSS hsl color property</title>
      <style>
          h1{
             text-align:center;
          }
         #rgb{
            color:rgb(255,0,0);
         }
          #rgba{
           color:rgba(255,0,0,0.5);
          }
         #hex{
            color:#EE82EE;
         }
          #short{
           color: #E8E;
          }
         #hsl{
            color:hsl(0,50%,50%);
         }
          #hsla{
           color:hsla(0,50%,50%,0.5);
          }
         #built{
            color:green;
         }
       </style>
   </head>
   <body>
      <h1 id="rgb">
           Hello World. This is RGB format.
      </h1>
       <h1 id="rgba">
        Hello World. This is RGBA format.
     </h1>
     <h1 id="hex">
      Hello World. This is Hexadecimal format.
  </h1>
  <h1 id="short">
  Hello World. This is Short-hexadecimal format.
</h1>
  <h1 id="hsl">
  Hello World. This is HSL format.
</h1>
<h1 id="hsla">
  Hello World. This is HSLA format.
</h1>
<h1 id="built">
  Hello World. This is Built-in color format.
</h1>
   </body>
</html>
```

There are three different formats to define a color:

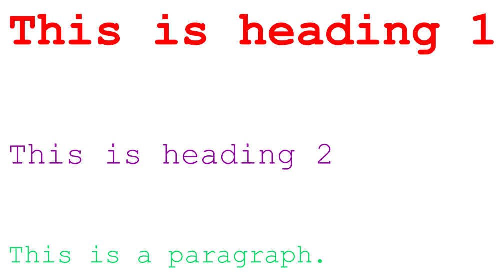
- By a color name
- By hexadecimal value
- By RGB

In the above example, we have defined all these formats.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   font-size: 100%;
}
h1 { color: red; }
h2 { color: #9000A1; }
p { color:rgb(0, 220, 98); }
}
</style>
</head>
```

```
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:



This is heading 1

This is heading 2

This is a paragraph.

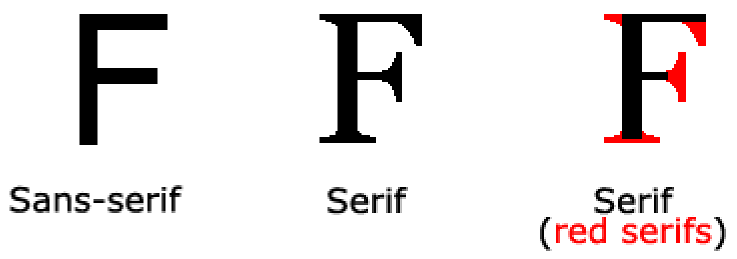
2) CSS Font Family

CSS font family can be divided in two types:

- Generic family: It includes Serif, Sans-serif, and Monospace.
- Font family: It specifies the font family name like Arial, New Times Roman etc.

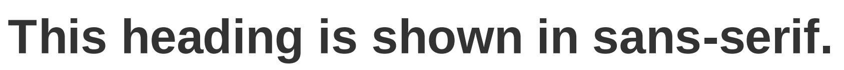
Serif: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

Sans-serif: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.

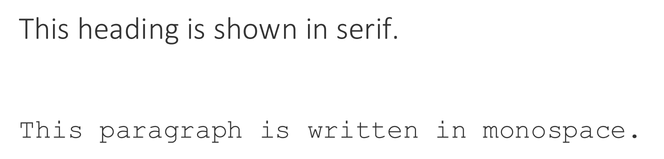


```
<!DOCTYPE html>
<html>
<head>
<style>
body {
font-size: 100%;
}
h1 { font-family: sans-serif; }
h2 { font-family: serif; }
p { font-family: monospace; }
}
</style>
</head>
<body>
<h1>This heading is shown in sans-serif.</h1>
<h2>This heading is shown in serif.</h2>
<p>This paragraph is written in monospace.</p>
</body>
</html>
```

Output:



This heading is shown in sans-serif.


This heading is shown in serif.

This paragraph is written in monospace.

3) CSS Font Size

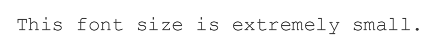
CSS font size property is used to change the size of the font.

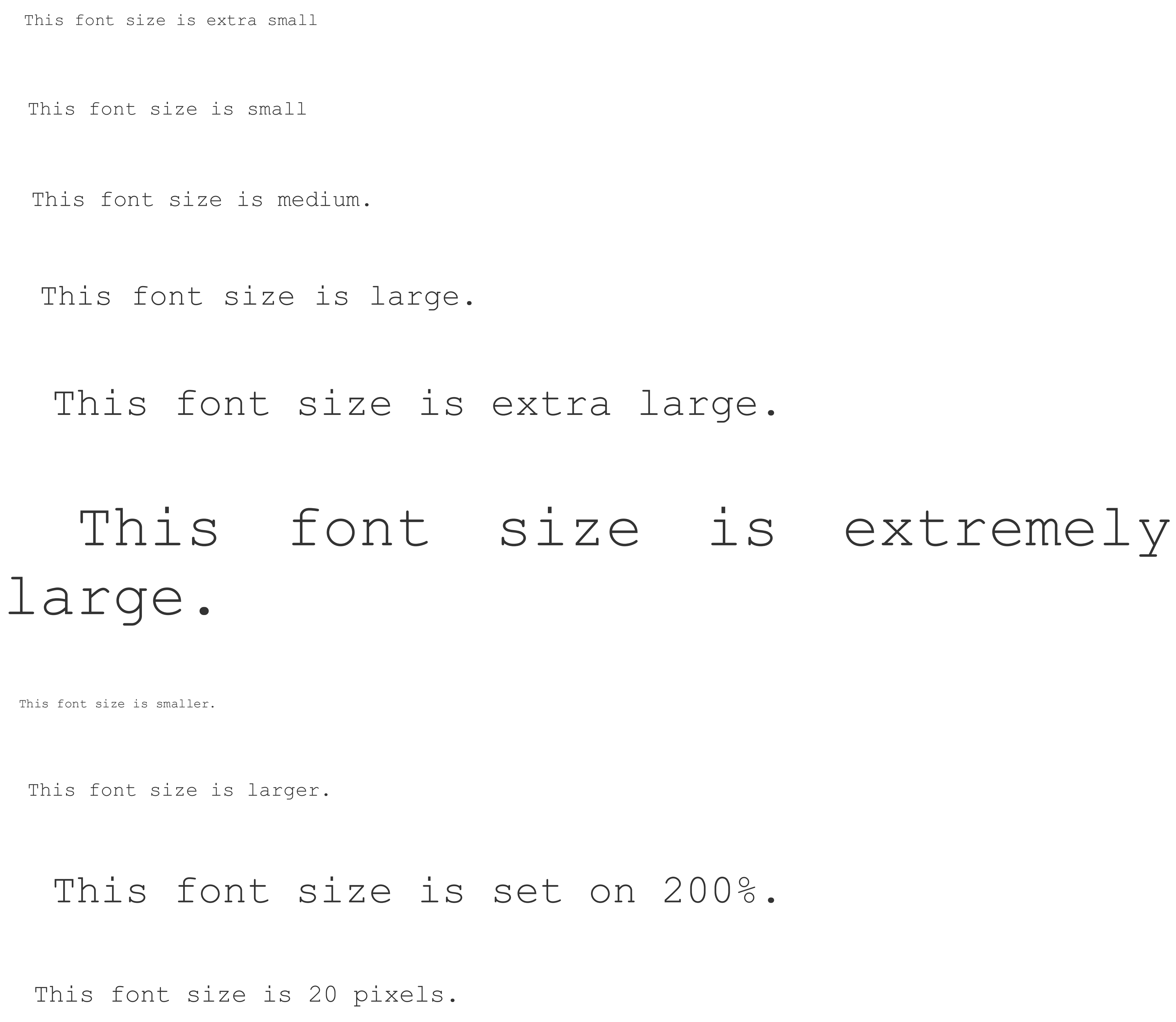
These are the possible values that can be used to set the font size:

Font Size Value	Description
xx-small	used to display the extremely small text size.
x-small	used to display the extra small text size.
small	used to display small text size.
medium	used to display medium text size.
large	used to display large text size.
x-large	used to display extra large text size.
xx-large	used to display extremely large text size.
smaller	used to display comparatively smaller text size.
larger	used to display comparatively larger text size.
size in pixels or %	used to set value in percentage or in pixels.

```
<html>
<head>
<title>Practice CSS font-size property</title>
</head>
<body>
<p style="font-size:xx-small;">  This font size is extremely small.</p>
<p style="font-size:x-small;">  This font size is extra small</p>
<p style="font-size:small;">  This font size is small</p>
<p style="font-size:medium;">  This font size is medium. </p>
<p style="font-size:large;">  This font size is large. </p>
<p style="font-size:x-large;">  This font size is extra large. </p>
<p style="font-size:xx-large;">  This font size is extremely large. </p>
<p style="font-size:smaller;">  This font size is smaller. </p>
<p style="font-size:larger;">  This font size is larger. </p>
<p style="font-size:200%;">  This font size is set on 200%. </p>
<p style="font-size:20px;">  This font size is 20 pixels.  </p>
</body>
</html>
```

Output:


This font size is extremely small.



```
This font size is extra small

This font size is small

This font size is medium.

This font size is large.

This font size is extra large.

This font size is extremely large.

This font size is smaller.

This font size is larger.

This font size is set on 200%.

This font size is 20 pixels.
```

4) CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
font-size: 100%;
}
h2 { font-style: italic; }
h3 { font-style: oblique; }
h4 { font-style: normal; }
}
</style>
</head>
<body>
<h2>This heading is shown in italic font.</h2>
<h3>This heading is shown in oblique font.</h3>
```

```
<h4>This heading is shown in normal font.</h4>
</body>
</html>
```

Output:

```
This heading is shown in italic font.

This heading is shown in oblique font.

This heading is shown in normal font.
```

5) CSS Font Variant

CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.

```
<!DOCTYPE html>
<html>
<head>
<style>
p { font-variant: small-caps; }
h3 { font-variant: normal; }
</style>
</head>
<body>
<h3>This heading is shown in normal font.</h3>
<p>This paragraph is shown in small font.</p>
</body>
</html>
```

Output:

```
This heading is shown in normal font.

THIS PARAGRAPH IS SHOWN IN SMALL FONT.
```

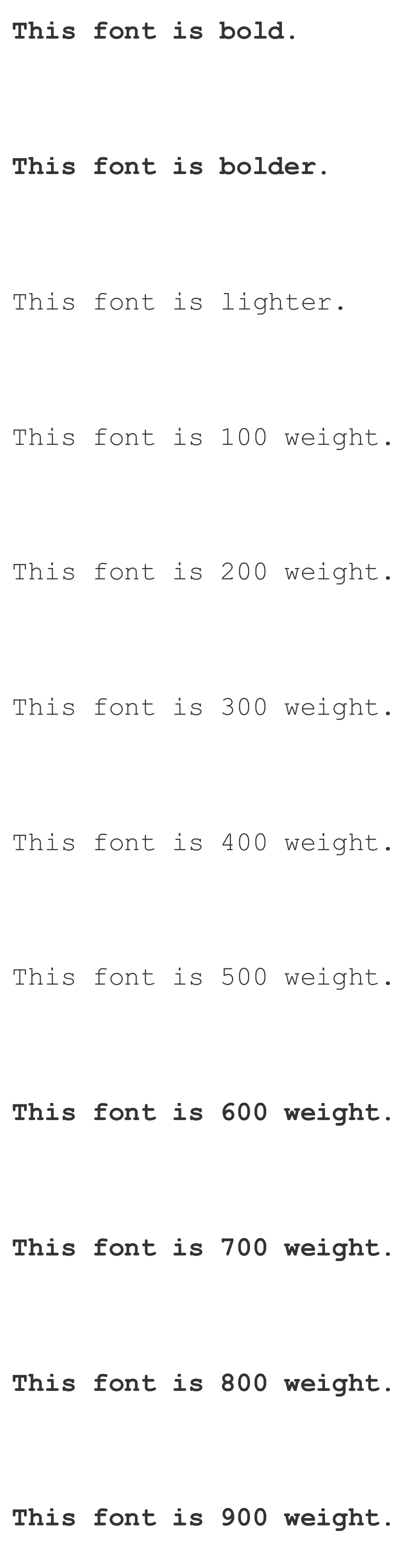
6) CSS Font Weight

CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).

```
<!DOCTYPE html>
<html>
<body>
<p style="font-weight:bold;">This font is bold.</p>
<p style="font-weight:bolder;">This font is bolder.</p>
<p style="font-weight:lighter;">This font is lighter.</p>
<p style="font-weight:100;">This font is 100 weight.</p>
<p style="font-weight:200;">This font is 200 weight.</p>
<p style="font-weight:300;">This font is 300 weight.</p>
<p style="font-weight:400;">This font is 400 weight.</p>
<p style="font-weight:500;">This font is 500 weight.</p>
<p style="font-weight:600;">This font is 600 weight.</p>
<p style="font-weight:700;">This font is 700 weight.</p>
<p style="font-weight:800;">This font is 800 weight.</p>
```

```
<p style="font-weight:900;">This font is 900 weight.</p>
</body>
</html>
```

Output:



```
This font is bold.

This font is bolder.

This font is lighter.

This font is 100 weight.

This font is 200 weight.

This font is 300 weight.

This font is 400 weight.

This font is 500 weight.

This font is 600 weight.

This font is 700 weight.

This font is 800 weight.

This font is 900 weight.
```

CSS Font-size

The font-size property in CSS is used to specify the height and size of the font. It affects the size of the text of an element. Its default value is medium and can be applied to every element. The values of this property include **xx-small**, **small**, **x-small**, etc.

Syntax

```
font-size: medium|large|x-large|xx-large|xx-small|x-small|small|;
```

The font-size can be relative or absolute.

Absolute-size

It is used to set the text to a definite size. Using absolute-size, it is not possible to change the size of the text in all browsers. It is advantageous when we know the physical size of the output.

Relative-size

It is used to set the size of the text relative to its neighboring elements. With relative-size, it is possible to change the size of the text in browsers.

NOTE: If we do not define a font-size, then for the normal text such as paragraphs, the default size is 16px, which is equal to 1em.

Font-size with pixels

When we set the size of text with pixels, then it provides us the full control over the size of the text.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>

#first {
  font-size: 40px;
}
#second {
  font-size: 20px;
}
</style>
</head>
<body>

<p id="first">This is a paragraph having size 40px.</p>
<p id="second">This is another paragraph having size 20px.</p>

</body>
</html>
```

Font-size with em

It is used to resize the text. Most of the developers prefer **em** instead of **pixels**. It is recommended by the world wide web consortium (W3C). As stated above, the default text size in browsers is 16px. So, we can say that the default size of **1em** is **16px**.

The formula for calculating the size from **pixels** to **em** is **em = pixels/16**.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
#first {
  font-size: 2.5em; /* 40px/16=2.5em */
}

#second {
  font-size: 1.875em; /* 30px/116=1.875em */
 }

#third {
  font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<body>

<p id='first'>First paragraph.</p>
<p id='second'>Second paragraph</p>
<p id='third'>Third Paragraph.</p>
```

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

Responsive font size

We can set the size of the text by using a **vw unit**, which stands for the '**viewport width**'. The viewport is the size of the browser window.

1vw = 1% of viewport width.

If the width of the viewport is 50cm, then the 1vw is equal to 0.5 cm.

Example

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


<p style="font-size:5vw;">First paragraph having the width of 5vw.</p>
<p style="font-size:10vw;">Second paragraph having the width of 10vw.</p>

</body>
</html>
```

Font-size with the length property

It is used to set the size of the font in length. The length can be in cm, px, pt, etc. Negative values of **length** property are not allowed in font-size.

Syntax

```
font-size: length;
```

Example

```
<!DOCTYPE html>
<html>
    <head>
      <style>
         .length {
            color:red;
             font-size: 5cm;
        }
       </style>
   </head>

   <body>
       <h1>font-size property</h1>

       <p class = "length">A paragraph having length 5cm.</p>
   </body>
</html>
```

CSS Background-color

This property is used to set the background color of an element. The background of an element covers the total size, including the padding and border and excluding margin. It can be applied to all HTML elements.

Syntax

```
element {
   background-color: color_name|transparent|initial|inherit;
}
```

Let's discuss the possible values of this property.

- **color_name:** It is used for defining the background color value or the color codes. It can be given by using the color name, hexadecimal value, or rgb() value.
- **transparent:** It is the default value of this property, which is used to specify the transparent background-color.
- **initial:** It is not used to set the background color. It sets the default value.
- **Inherit:** It is used to inherit the background-color from its parent.

Let's see an illustration of this property.

Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>background-color property</title>
       <style>
          body {
              text-align:center;
             background-color: lightblue;
           }
       h1{
          color: blue;
       }
       </style>
   </head>
   <body>
     <h1>Hello World.</h1>
       <h1>Welcome to the javaTpoint.com</h1>
      </body>
</html>
```

CSS Margin

CSS Margin property is used to define the space around elements. It is completely transparent and doesn't have any background color. It clears an area around the element.

Top, bottom, left and right margin can be changed independently using separate properties. You can also change all properties at once by using shorthand margin property.

There are following CSS margin properties:

CSS Margin Properties

Property	Description
margin	This property is used to set all the properties in one declaration.
margin-left	it is used to set left margin of an element.
margin-right	It is used to set right margin of an element.

margin-top	It is used to set top margin of an element.
margin-bottom	It is used to set bottom margin of an element.

CSS Margin Values

These are some possible values for margin property.

Value	Description
auto	This is used to let the browser calculate a margin.
length	It is used to specify a margin pt, px, cm, etc. its default value is 0px.
%	It is used to define a margin in percent of the width of containing element.
inherit	It is used to inherit margin from parent element.

Note: You can also use negative values to overlap content.

CSS margin Example

You can define different margin for different sides for an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.ex {
    margin-top: 50px;
    margin-bottom: 50px;
    margin-right: 100px;
    margin-left: 100px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
</body>
</html>
```

Output:

```
This paragraph is not displayed with specified margin.


          This paragraph is displayed with specified margin.
```

Margin: Shorthand Property

CSS shorthand property is used to shorten the code. It specifies all the margin properties in one property.

There are four types to specify the margin property. You can use one of them.

1. margin: 50px 100px 150px 200px;
2. margin: 50px 100px 150px;
3. margin: 50px 100px;
4. margin 50px;

1) margin: 50px 100px 150px 200px;

It identifies that:

top margin value is 50px

right margin value is 100px

bottom margin value is 150px

left margin value is 200px

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.ex {
     margin: 50px 100px 150px 200px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
</body>
</html>
```

Output:

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

2) margin: 50px 100px 150px;

It identifies that:

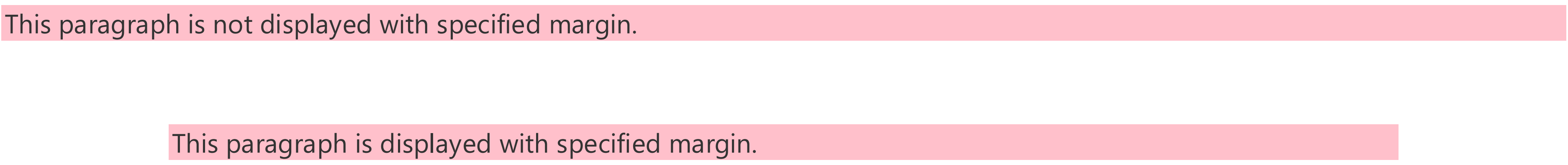
top margin value is 50px

left and right margin values are 100px

bottom margin value is 150px

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.ex {
    margin: 50px 100px 150px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
</body>
</html>
```

Output:



This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

3) margin: 50px 100px;

It identifies that:

top and bottom margin values are 50px

left and right margin values are 100px

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.ex {
    margin: 50px 100px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
```

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

Output:

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

4) margin: 50px;

It identifies that:

top right bottom and left margin values are 50px

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.ex {
    margin: 50px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
</body>
</html>
```

Output:

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

CSS Padding

CSS Padding property is used *to define the space between the element content and the element border*.

It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.

Top, bottom, left and right padding can be changed independently using separate properties. You can also change all properties at once by using shorthand padding property.

CSS Padding Properties

Property	Description

padding	It is used to set all the padding properties in one declaration.
padding-left	It is used to set left padding of an element.
padding-right	It is used to set right padding of an element.
padding-top	It is used to set top padding of an element.
padding-bottom	It is used to set bottom padding of an element.

CSS Padding Values

Value	Description
length	It is used to define fixed padding in pt, px, em etc.
%	It defines padding in % of containing element.

CSS Padding Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    background-color: pink;
}
p.padding {
    padding-top: 50px;
    padding-right: 100px;
    padding-bottom: 150px;
    padding-left: 200px;
}
</style>
</head>
<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>
</html>
```

Output:

```
This is a paragraph with no specified padding.

This is a paragraph with specified paddings.
```

CSS Width

The **CSS width property** is used *to set the width of the content area of an element*.

It does not include padding borders or margins. It sets width of the area inside the padding, border, and margin of the element.

CSS width values

Value	Description

auto	It is a default value. it is used to calculate the width.
length	It is used to define the width in px, cm etc.
%	It defines the width of the containing block in %.
initial	It is used to set the property to its default value.
inherit	It is used to inherit the property from its parent element.

CSS Width Example: width in px

```
<!DOCTYPE html>
<html>
<head>
<style>
img.normal {
    width: auto;
}
img.big {
    width: 150px;
}
p.ex {
    height: 150px;
    width: 150px;
}
</style>
</head>
<body>
<img class="normal" src="good-morning.jpg" width="95" height="84"><br>
<img class="big" src="good-morning.jpg" width="95" height="84">
<p class="ex">The height and width of this paragraph is 150px.</p>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:

The height and width of this paragraph is 150px.

This is a paragraph.

CSS Width Example: width in %

The percent width is a measurement unit for the containing block. It is great for images.

```
<!DOCTYPE html>
<html>
```

```
<head>
<style>
img.normal {
    width: auto;
}
img.big {
    width: 50%;
}
img.small {
    width: 10%;
}
</style>
</head>
<body>
<img class="normal" src="good-morning.jpg" width="95" height="84"><br>
<img class="big" src="good-morning.jpg" width="95" height="84"><br>
<img class="small" src="good-morning.jpg" width="95" height="84">
</body>
</html>
```

Output:

Note: You can also use the "min-width" and "max-width" property to control the size of image.

CSS height property

This CSS property sets the height of an element. It is used to set the height of content area of an element.

It does not include padding borders or margins, whereas it sets the height of the area inside the padding, border, and margin of the element. It can accept the length and percentage values. But it does not allow negative values.

If we set the height to a numeric value (like in px, %, etc.), the content can be overflow if it does not fit in the given height. We can manage the overflowing content by defining the **overflow** property.

If the height of the container is not explicitly defined, and the element is not absolutely positioned (i.e., **position: absolute;**), the value of **height** property is set to **auto**. The min-height and max-height properties can also be used to control the size.

Syntax

```
height: auto | length | initial | inherit;
```

Property Values

The values of this property are tabulated as follows.

Value	Description
auto	It is a default value. Using this value browser is responsible for calculating the height of the element. Negative values are not allowed.

length	It specifies the height of an element using the length units such as px, cm, pt, etc. Negative values are not allowed.
%	It defines the height of the container in %. Negative values are not allowed.
initial	It is used to set the property to its default value.
inherit	It is used to inherit the property from its parent element.

Now, we will see some of the examples to understand this property more clearly.

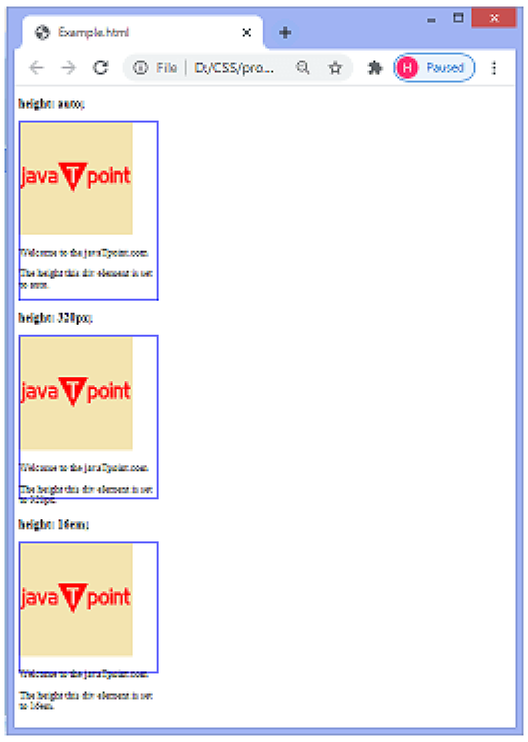
Example

Here, we are using the keyword value **auto** and the length values of height property in **px** and **em**.

```
<!DOCTYPE html>
<html>
<head>
<style>
#auto{
height: auto;
width: 275px;
border: 2px solid blue;
}
#px{
height: 320px;
width: 275px;
border: 2px solid blue;
}
#em{
height: 16em;
width: 275px;
border: 2px solid blue;
}
p{
font-size: 20px;
}
</style>
</head>
<body>
<h2> height: auto; </h2>
<div id ="auto">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to auto. </p>
</div>
<h2> height: 320px; </h2>
<div id ="px">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to 320px. </p>
</div><br>
<h2> height: 16em; </h2>
<div id ="em">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
```

```
<p> The height this div element is set to 16em. </p>
</div>
</body>
</html>
```

Output



height: auto;

Welcome to the javaTpoint.com

The height this div element is set to auto.

height: 320px;

Welcome to the javaTpoint.com

The height this div element is set to 320px.

height: 16em;

Welcome to the javaTpoint.com

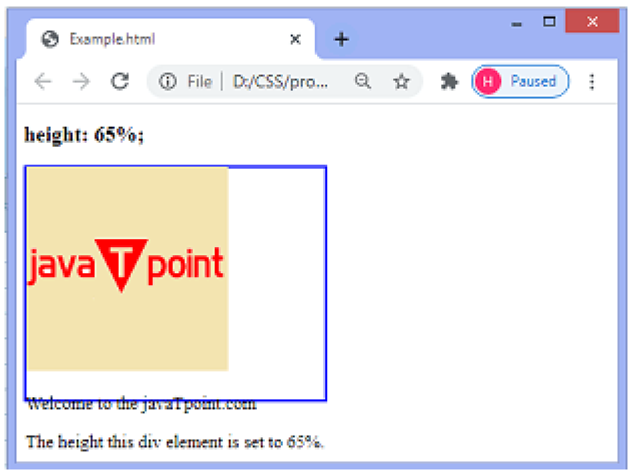
The height this div element is set to 16em.

Example

Here, we are specifying the value of **height** property in percentage.

```
<!DOCTYPE html>
<html>
<head>
<style>
#per{
position: absolute;
width: auto;
height: 65%;
border: 2px solid blue;
}

p{
font-size: 20px;
}

</style>
</head>
<body>

<h2> height: 65%; </h2>
<div id ="per">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to 65%. </p>
</div>
</body>
</html>
```

Output



height: 65%;

Welcome to the javaTpoint.com

The height this div element is set to 65%.

Box-shadow CSS

It is used to add shadow-like effects around the frame of an element.

Syntax

```
box-shadow: h-offset v-offset blur spread color |inset|inherit|initial|none;
```

Let's understand property values.

h-offset: It horizontally sets the shadow position. Its positive value will set the shadow to the right side of the box. Its negative value is used to set the shadow on the left side of the box.

v-offset: Unlike the **h-offset**, it is used to set the shadow position vertically. The positive value in it sets the shadow below the box, and the negative value sets the shadow above of the box.

blur: As its name implies, it is used to blur the box-shadow. This attribute is optional.

spread: It sets the shadow size. The spread size depends upon the spread value.

color: As its name implies, this attribute is used to set the color of the shadow. It is an optional attribute.

inset: Normally, the shadow generates outside of the box, but by using inset, the shadow can be created within the box.

initial: It is used to set the property of the box-shadow to its default value.

inherit: it is inherited from its parent.

none: It is the default value that does not include any shadow property.

Let's understand the above attributes by using an illustration.

Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>CSS box-shadow Property</title>

      <style>
          div
         {
          border: 1px solid;
         padding: 10px;
          }
         #hvb
          {
            /* box-shadow: h-offset v-offset blur */
             box-shadow: 5px 10px 10px;
         }
          #spr
         {
             /* box-shadow: h-offset v-offset blur spread */
            box-shadow: 5px 10px 10px 10px;
          }
         #col
          {
            /* box-shadow: h-offset v-offset blur spread color */
             box-shadow: 5px 10px 10px 10px orange;
         }
          #ins
         {
              /* box-shadow: h-offset v-offset blur spread color inset */
            box-shadow: 5px 10px 10px 10px orange inset;
          }
         #init
          {
             /* box-shadow: initial */
            box-shadow: initial;

          }
         #non
          {
            /* box-shadow: none */
             box-shadow: none;
         }
     </style>
   </head>

   <body>
     <div id = "hvb">
         <h1>It is a shadow box that has h-offset, v-offset and blur attributes.</h1>
      </div>
<br><br>
      <div id = "spr">
         <h1>It is a box that includes the spread attribute.</h1>
      </div>
<br><br>
      <div id = "col">
      <h1>It is a box that includes the color attribute.</h1>
   </div>
<br><br>
   <div id = "ins">
      <h1>It is a box that includes the inset attribute.</h1>
   </div>
<br><br>
   <div id = "init">
      <h1>It is a box that includes the initial attribute.</h1>
   </div>
<br><br>
   <div id = "non">
      <h1>It is a box that includes the default attribute i.e. none.</h1>
   </div>
   </body>
</html>
```

CSS text-align

This CSS property is used to set the horizontal alignment of a table-cell box or the block element. It is similar to the **vertical-align** property but in the horizontal direction.

Syntax

```
text-align: justify | center | right | left | initial | inherit;
```

Possible values

justify: It is generally used in newspapers and magazines. It stretches the element's content in order to display the equal width of every line.

center: It centers the inline text.

right: It is used to align the text to the right.

left: It is used to align the text to the left.

Let's see an example that will demonstrate the **text-align** property.

Example

```
<html>
  <head>
  </head>
<style>
h2{
color: blue;
}
</style>
  <body>
  <h1>Example of text-align proeprty</h1>

  <h2 style = "text-align: center;">
       text-align: center;
    </h2>

    <h2 style = "text-align: right;">
       text-align: right;
    </h2>

    <h2 style = "text-align: left;">
        text-align: left;
    </h2>
    <h2 style = "text-align: justify;">
       text-align: justify; To see its effect, it should be applied on large paragraph.
    </h2>

    </body>
</html>
```

CSS Animation

CSS Animation property is used *to create animation on the webpage*. It can be used as a replacement of animation created by Flash and JavaScript.

CSS3 @keyframes Rule

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a CSS animation sequence.

What animation does

An animation *makes an element change gradually* from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage.0% specify the start of the animation and 100% specify its completion.

How CSS animation works

When the animation is created in the @keyframe rule, it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

- The name of the animation
- The duration of the animation

CSS animation properties

Property	Description
@keyframes	It is used to specify the animation.
animation	This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property.
animation-delay	It specifies when the animation will start.
animation-direction	It specifies if or not the animation should play in reserve on alternate cycle.
animation-duration	It specifies the time duration taken by the animation to complete one cycle.
animation-fill-mode	it specifies the static style of the element. (when the animation is not playing)
animation-iteration-count	It specifies the number of times the animation should be played.
animation-play-state	It specifies if the animation is running or paused.
animation-name	It specifies the name of @keyframes animation.
animation-timing-function	It specifies the speed curve of the animation.

CSS animation example: changing background color

Let's see a simple CSS animation example that changes background color of rectangle from RED to BLACK.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-animation: myfirst 6s; /* Chrome, Safari, Opera */
    animation: myfirst 5s;
}
/* Chrome, Safari, Opera */
@-webkit-keyframes myfirst {
    from {background: red;}
    to {background: green;}
}
/* Standard syntax */
```

```
@keyframes myfirst {
    from {background: red;}
    to {background: green;}
}
</style>
</head>
<body>
<p><b>Note:</b> The IE 9 and earlier versions don't support CSS3 animation property. </p>
<div></div>
</body>
</html>
```

CSS animation example: Moving Rectangle

Let's take another example to display animation with percentage value.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    -webkit-animation: myfirst 5s; /* Chrome, Safari, Opera */
    animation: myfirst 5s;
}
/* Chrome, Safari, Opera */
@-webkit-keyframes myfirst {
    0%   {background:red; left:0px; top:0px;}
    25%  {background:yellow; left:300px; top:0px;}
    50%  {background:blue; left:200px; top:300px;}
    75%  {background:green; left:0px; top:200px;}
    100% {background:red; left:0px; top:0px;}
}
/* Standard syntax */
@keyframes myfirst {
    0%   {background:red; left:0px; top:0px;}
    25%  {background:yellow; left:300px; top:0px;}
    50%  {background:blue; left:300px; top:200px;}
    75%  {background:green; left:0px; top:200px;}
    100% {background:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<p><b>Note:</b> The Internet Explorer 9 and its earlier versions don't support this example.</p>
<div></div>
</body>
</html>
```

CSS Grid

A grid can be defined as an intersecting set of horizontal lines and vertical lines. CSS Grid layout divides a page into major regions. It defines the relationship between the parts of a control built from HTML primitives in terms of layer, position, and size. Grid property offers a grid-based layout system having rows and columns. It makes the designing of web pages easy without positioning and floating.

Similar to the table, it enables a user to align the elements into rows and columns. But compare to tables, it is easy to design layout with the CSS grid. We can define columns and rows on the grid by using **grid-template-rows** and **grid-template-columns** properties.

The CSS **grid property** is supported in browsers such as **Google Chrome, Internet Explorer, Firefox, Safari,** and **Opera**.

Grid Container

We can define the grid container by setting the **display** property to **grid** or **inline-grid** on an element.

Grid container contains grid items that are placed inside rows and columns.

Let's see a simple example of a grid in CSS.

Example

```
<!DOCTYPE html>
<html>

<head>

  <style>
    .main {
      display: grid;
       grid: auto auto / auto auto auto auto;
      grid-gap: 10px;
       background-color: black;
      padding: 10px;
    }

    .num {
      background-color: grey;
       text-align: center;
      color: white;
       padding: 10px 10px;
      font-size: 30px;
    }
  </style>
</head>

<body>
  <div class="main">
     <div class="num">One</div>
    <div class="num">Two</div>
     <div class="num">Three</div>
    <div class="num">Four</div>
     <div class="num">Five</div>
    <div class="num">Six</div>
     <div class="num">Seven</div>
    <div class="num">Eight</div>
   </div>

</body>

</html>
```

Let's see some of the shorthand properties:

- **grid-template-columns:** It is used to specify the size of the columns.
- **grid-template-rows:** It is used to specify the row size.
- **grid-template-areas:** It is used to specify the grid layout by using the named items.
- **grid-auto-rows:** It is used to specify the automatic size of the rows.
- **grid-auto-columns:** It is used to specify the automatic size of the columns.
- **grid-auto-flow:** It is used to specify how to place auto-placed items and the automatic row size.

In the following example, we are including some of the above shorthand properties. Now, let's see the example of using some properties:

Example

```
<!DOCTYPE html>
<html>

<head>

  <style>
     .main {
        display: grid;
         grid-template-columns: 1fr 1fr 1fr;
        grid-template-rows: 100px 250px 200px;
         background-color: black;
        grid-gap: 10px;
         padding: 20px;
     }

    .num {
         background-color: lightgrey;
        text-align: center;
         padding: 20px 10px;
        font-size: 30px;
     }
  </style>
</head>

<body>

  <div class="main">
     <div class="num">One</div>
      <div class="num">Two</div>
     <div class="num">Three</div>
      <div class="num">Four</div>
     <div class="num">Five</div>
      <div class="num">Six</div>
     <div class="num">Seven</div>
      <div class="num">Eight</div>
  </div>

</body>

</html>
```

The justify-content property

It is used to align the entire grid within the container. It includes values such as:

- **space-evenly:** It provides equal space in between or around the columns.
- **space-around:** It provides equal space around the columns.
- **space-between:** It gives an equal amount of space between the columns.
- **center:** It is used to align the grid in the middle of the container.
- **start:** It is used to align the grid at the beginning of the container.
- **end:** It is used to align the grid at the end of the container.

Note: It is noted that the total width of the grid should be less than the width of the container for any effect of the justify-content property.

Let's understand this property along with the values by using an example.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container1 {
  display: grid;
  justify-content: space-evenly;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container2 {
  display: grid;
  justify-content: space-around;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: red;
  padding: 10px;
}

.grid-container3 {
  display: grid;
  justify-content: space-between;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: green;
  padding: 10px;
}

.grid-container4 {
  display: grid;
  justify-content: end;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: violet;
  padding: 10px;
}

.grid-container5 {
  display: grid;
  justify-content: start;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: gray;
  padding: 10px;
}

.grid-container6 {
  display: grid;
  justify-content: center;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than the container*/
  grid-gap: 10px;
  background-color: blue;
  padding: 10px;
}

.num {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
</head>
<body>

   <b>CONTAINER WITH SPACE-EVENLY VALUE</b>
<div class="grid-container1">
  <div class='num'>1</div>
  <div class='num'>2</div>
  <div class='num'>3</div>
</div>

<b>CONTAINER WITH SPACE-AROUND VALUE</b>
<div class="grid-container2">
   <div class='num'>1</div>
   <div class='num'>2</div>
   <div class='num'>3</div>
 </div>

 <b>CONTAINER WITH SPACE-BETWEEN VALUE</b>
<div class="grid-container3">
   <div class='num'>1</div>
   <div class='num'>2</div>
   <div class='num'>3</div>
 </div>

 <b>CONTAINER WITH END VALUE</b>
 <div class="grid-container4">
    <div class='num'>1</div>
    <div class='num'>2</div>
    <div class='num'>3</div>
  </div>

  <b>CONTAINER WITH START VALUE</b>
<div class="grid-container5">
```

```
<div class='num'>1</div>
  <div class='num'>2</div>
  <div class='num'>3</div>
 </div>

 <b>CONTAINER WITH CENTER VALUE</b>
<div class="grid-container6">
  <div class='num'>1</div>
  <div class='num'>2</div>
  <div class='num'>3</div>
 </div>

</body>
</html>
```

The align-content property

It is used to align the entire grid within the container vertically.

> ***Note: It is noted that the total height of the grid should be less than the height of the container for any effect of the align-content property.***

The values of the **align-content** property are the same as the values of the **justify-content** property.

CSS Layout

CSS layout is easy to design. We can use CSS layout to design our web page such as home page, contact us, about us etc.

There are 3 ways to design layout of a web page:

1. **HTML Div with CSS**: fast and widely used now.
2. **HTML Table**: slow and less preferred.
3. **HTML Frameset**: deprecated now.

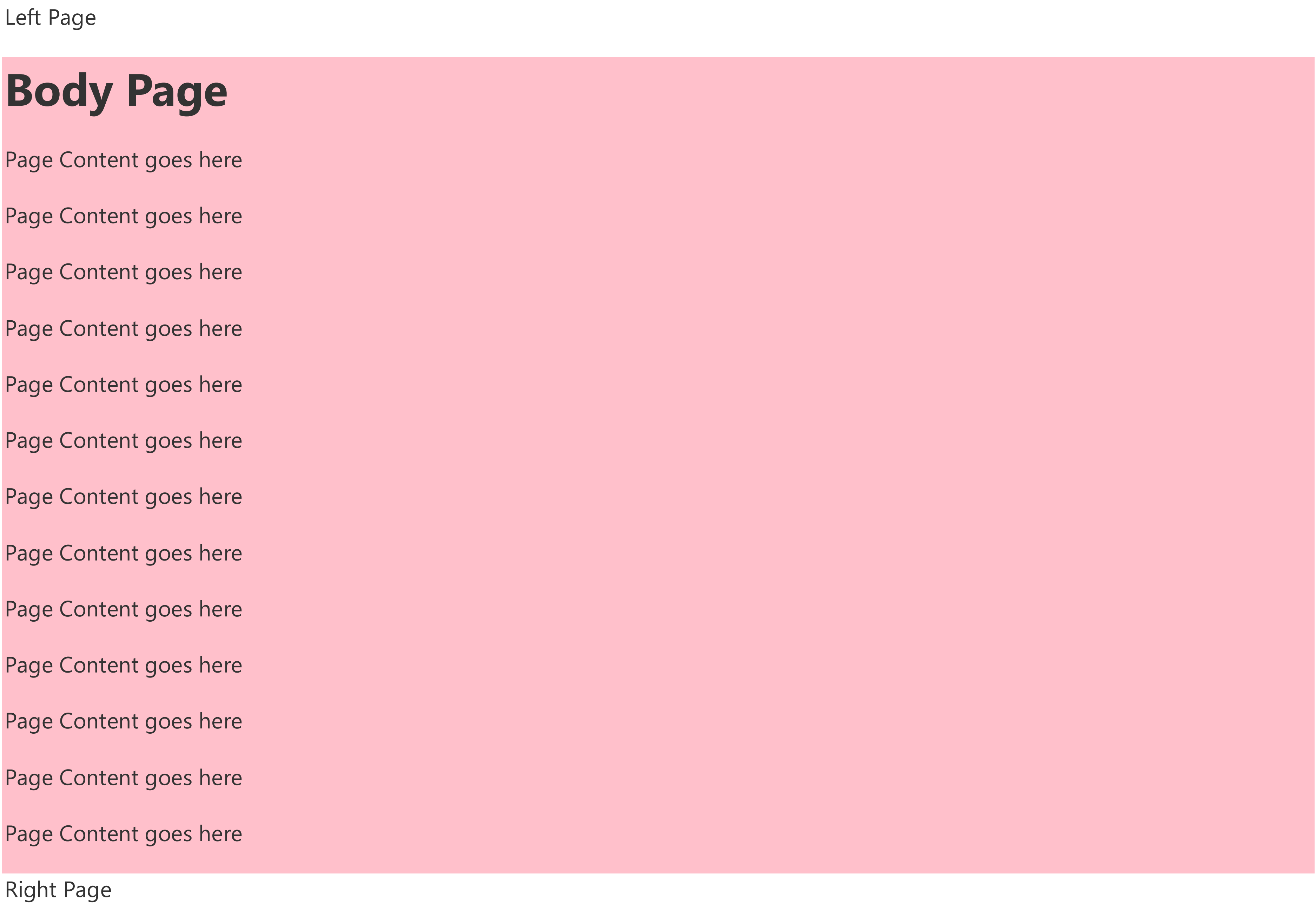
A CSS layout can have header, footer, left pane, right pane and body part. Let's see a simple example of CSS layout.

CSS layout example

```
<!DOCTYPE html>
<html>
<head>
<style>
.header{margin:-8px -8px 0px;background-image:linear-gradient(145deg,#7379ff,#b524ef);color:white;text-align:center;padding:10px;}
.container{width:100%}
.left{width:15%;float:left;}
.body{width:65%;float:left;background-color:pink;padding:5px;}
.right{width:15%;float:left;}
.footer{margin:-8px;clear:both;background-image:linear-gradient(145deg,#7379ff,#b524ef);color:white;text-align:center;padding:10px;}
</style>
</head>
<body>
<div class="header"><h2>JavaTpoint</h2></div>

<div class="container">
<div class="left">
```

```
<p>Left Page</p>
</div>
<div class="body">
<h1>Body Page</h1>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p>
</div>
<div class="right">
<p>Right Page</p>
</div>
</div>

<div class="footer">
<p>Footer</p>
</div>

</body>
</html>
```

Output:



```
Left Page

Body Page

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Page Content goes here

Right Page
```

CSS Table

We can apply style on HTML tables for better look and feel. There are some CSS properties that are widely used in designing table using CSS:

- border
- border-collapse
- padding
- width
- height
- text-align
- color
- background-color

CSS Table Border

We can set border for the table, th and td tags using the CSS border property.

```
<style>
table, th, td {
    border: 1px solid black;
}
</style>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

CSS Table Border Collapse

By the help of border-collapse property, we can collapse all borders in one border only.

```
<style>
table, th, td {
    border: 2px solid black;
    border-collapse: collapse;
}
</style>
```

Output:

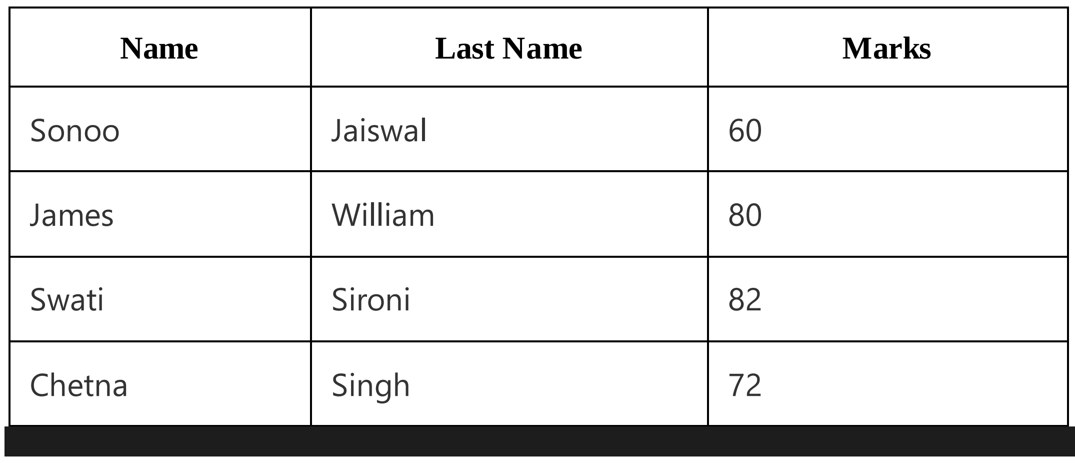
Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

CSS Table Padding

We can specify padding for table header and table data using the CSS padding property.

```
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 10px;
}
</style>
```

Output:



Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

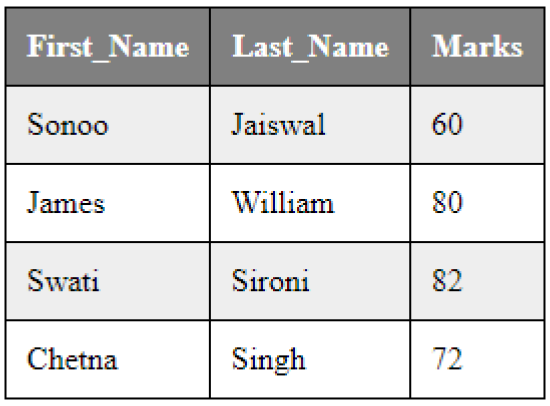
CSS Table: Styling even and odd cells

We can style even and odd table cells for better look and feel. In this code, we are displaying different background colors on even and odd cells. Moreover, we have changed the background-color and color of <th> tag.

CSS code:

```
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 10px;
}
table#alter tr:nth-child(even) {
    background-color: #eee;
}
table#alter tr:nth-child(odd) {
    background-color: #fff;
}
table#alter th {
    color: white;
    background-color: gray;
}
</style>
```

Output:



First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72