[**CSS**](https://www.geeksforgeeks.org/css/)**:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages.

Let us breakdown the word CSS or Cascading Style Sheet

**Cascading:** means that styles can fall (or cascade) from one style sheet to another. That is, there can be multiple CSS files linked to the same HTML document, where the last one matters the most.

**Style:** Adding Styling our HTML tags

**Sheets:** Writing our style code in different documents

CSS is designed to make style sheets for web pages. It is independent, i.e., it can be used with any XML-based markup language.

CSS is designed to make style sheets for web pages. It is independent, i.e., it can be used with any XML-based markup language.

**Cascading Style Sheet (CSS) Syntax**

Selector {

Property\_1 : value;

Property\_2 : value;

Property\_3 : value;

}

**Example:**

h1{

Text-align: left;

color: blue;

}

**Importing CSS to HTML:**

There are, in total, three ways of inserting CSS into an HTML page. The ways are :

* Inline CSS
* Internal CSS
* External CSS

**Inline CSS:**

It is a way of adding a unique style to a particular element.

To use Inline Styles, you can add the style attribute in the relevant HTML tag, and then inside the style attribute, you can provide different styles.

**Syntax:**

<element style="property:value; property:value;"></element>

**Internal CSS:**

Internal CSS is used under one single HTML page. For inserting internal CSS, we place the whole CSS style inside the head tag of HTML by using the <style> tag.

**Example:**

<head>

<style>

body {

background-color: linen;

}

h1 {

color: Black;

text-align: center;

}

h2 {

color: #ff69b4;

text-align: center;

}

</style>

</head>

<body>

<h1>Header One</h1>

<h2>Header Two</h2>

</body>

</html>

**External CSS :**

But, inserting **CSS inline or internal** or internal are , **not efficient**. What if we want to use the **same CSS for multiple web pages**? If we use the inline or internal insertion method, there will be a **lot of redundancy**. So, to avoid that issue, we have **external CSS**.

Here, we use the same **CSS** file again and again. The way we do it is by making a **CSS** file and linking it to the **HTML** inside the <head> tag using the <link> tag.

**Syntax :**

<link rel="stylesheet" href="file\_name.css">

Here, in href, we provide the path of the **CSS** file we want to use for our **HTML** Page. **Here in href, we provide the path of the CSS file we want to use for our HTML Page.**

**Example:**

<head>

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

</head>

<body>

<h1>Header One</h1>

<h2>Header Two</h2>

</body>

**file\_name.css :**

body{

background-color: linen;

}

h1 {

color: Black;

text-align: center;

}

h2 {

color: #FF69B4;

text-align: center;

}

**CSS Color:**

CSS Color property is used to set the color of HTML elements. This property is used to set font color, background color etc.

Color of an element can be defined in the following ways:

* Built-In Color
* RGB Format
* RGBA Format
* Hexadecimal Notation
* HSL
* HSLA

**Built-In Color:** These are a set of predefined colors which are used by its name. For example: red, blue, green etc.

**Syntax:**

h1 {

color: color-name;

}

**RGB Format:** The RGB(Red, Green, Blue) format is used to define the color of an HTML element by specifying the R, G, B values range between 0 to 255. For example: RGB value of Red color is (255, 0, 0), Green color is (0, 255, 0), Blue color is (0, 0, 255) etc.  
**Syntax:**

h1 {

color: rgb(R, G, B);

}

**RGBA Format:** The RGBA format is similar to the RGB, but the difference is RGBA contains A (Alpha) which specify the transparency of elements. The value of alpha lies between 0.0 to 1.0 where 0.0. represents fully transparent and 1.0 represents not transparent.  
**Syntax:**

h1 {

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

}

**Hexadecimal Notation:** The hexadecimal notation begins with # symbol followed by 6 characters each range from 0 to F. For example: Red #FF0000, Green #00FF00, Blue #0000FF etc.  
**Syntax:**

h1 {

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

}

**HSL:** HSL stands for Hue, Saturation, and Lightness respectively. This format uses the cylindrical coordinate system.

* **Hue:** Hue is the degree of the color wheel. Its value lies between 0 to 360 where 0 represents red, 120 represents green and 240 represents blue color.
* **Saturation:** It takes percentage value, where 100% represents completely saturated, while 0% represents completely unsaturated (gray).
* **Lightness:** It takes percentage value, where 100% represents white, while 0% represents black.

**Text Color:** It is used to set the color of the text.  
**Syntax:**

h1 {

color:color\_name;

}

**Background Color:** It is used to set the background color of an HTML element.  
**Syntax:**

h1 {

background-color:color\_name;

}

**Border Color:** It is used to set the border color of an element. Border-style is used to set the border type.  
**Syntax:**

h1 {

border-style:solid/dashed/dotted

border-color:color\_name;

}

**CSS Text Formatting:**

The CSS **text formatting** properties are used to format text, style the text and perform different types of manipulations like word spacing, alignment, justification, and text transformation.

**Syntax:**

The Syntax to write this property:

Selector {

property-name : /\*value\*/

}

### ****CSS Text Formatting Properties:****

These are the following text formatting properties:

* [**color**](https://www.geeksforgeeks.org/css-colors/): This property is used to set the color of the text and the color can be set by using a color name like “red”, hex value “#ff0000”, or by its RGB value “rgb(255,0,0)”;
* [**text-align**](https://www.geeksforgeeks.org/css-text-align-last-property/): This property in CSS is used to specify the horizontal alignment of text in an element inside a block element or table-cell box.
* [**text-align-last**](https://www.geeksforgeeks.org/css-text-align-last-property/)**:** It is used to set the last line of the paragraph just before the line break. It sets the alignment of all the last lines occurring in the element in which the text-align-last property is applied.
* [**text-decoration**](https://www.geeksforgeeks.org/css-text-decoration-property/)**:** text-decoration property is used to “decorate ” the content of the text.
* [**text-decoration-color**](https://www.geeksforgeeks.org/css-text-decoration-color-property/)**:** It is used to set the color of the decorations (overlines, underlines, and line-throughs) over the text.
* [**text-decoration-line**](https://www.geeksforgeeks.org/css-text-decoration-line-property/)**:** It is used to set the various kinds of text decorations. this may include many values such as underline, overline, line-through, etc.
* [**text-decoration-style**](https://www.geeksforgeeks.org/css-text-decoration-style-property/)**:** This property is used to set the text-decoration of the element. It is the combination of the text-decoration-line and text-decoration-color properties.
* [**text-indent**](https://www.geeksforgeeks.org/css-text-indent-property/)**:** It is used to indent the first line of the paragraph and the size can be in px, cm, pt.
* [**text-justify**](https://www.geeksforgeeks.org/css-text-justify-property/)**:** This property is used to set the text-align to justify. It spreads the words into complete lines.
* [**text-overflow**](https://www.geeksforgeeks.org/css-text-overflow-property/)**:** This property of text formatting specify that some text has overflown and is hidden from the view.
* [**text-transform**](https://www.geeksforgeeks.org/css-text-transform-property/)**:** It is used to control the capitalization of the text.
* [**text-shadow**](https://www.geeksforgeeks.org/css-text-shadow-property/)**:** it is used to add shadow to the text.
* [**letter-spacing**](https://www.geeksforgeeks.org/css-letter-spacing-property/)**:** This property is used to specify the space between the characters of the text.
* [**line-height**](https://www.geeksforgeeks.org/css-line-height-property/)**:** It is used to set the space between the lines.
* [**direction**](https://www.geeksforgeeks.org/css-direction-property/)**:** This property is used to set the direction of the text.
* [**word-spacing**](https://www.geeksforgeeks.org/css-word-spacing-property/)**:** It is used to specify the space between the words of the line.

**CSS Fonts:**

We all browse various websites each day, and a majority of them differ in their appearance. We find some of them exceptionally attractive and readable, while some of them are just mundane.

What makes these sites so distinguishable, and what leads to better usability and readability?

The answer is **"Fonts and Typography".**

**A simple HTML page can be made attractive by beautifying it with CSS. We can use various fonts, make them bold, change their color and adjust the font style and size accordingly. This can be done using font-family in CSS and font style in CSS.**

**Why is Font Selection Important?**

Fonts play a vital role in all websites. They not only make the text readable and eye-catching but also retain the user's attention. Who likes a bland website, after all?

It is important to use a font that the user can comprehend. Also, using the right size, color scheme, bold/italics, variant, and weight makes it interactive.

**What is Font Family in CSS?**

The font family in CSS is a set of fonts that have common designs and properties. The fonts which are defined in the font family differ in font weight like bold, italic, etc.

The font-family property is used in CSS to specify the font for a text. It is a prioritized list of various fonts, from the highest to lowest priority which is separated by commas.

They are prioritized at the user's discretion which he/she mentions, and then the browser checks for the font's compatibility on the device.

The browser selects the first font in the list and checks if that is installed or not, otherwise proceeds to the next font. Many browsers do not support all the fonts, so it's better to use multiple fonts.

This is known as **font-stack**, where we specify multiple fonts so that if the browser is not able to apply the first font on the result, it proceeds to the next font to apply it.

If it is not available either, this process continues till the next compatible font is encountered in the list that we have mentioned. It might happen that all the fonts that we have mentioned in the list are not compatible, so in this case, it is always recommended to use a generic family name at the end of the font list.

This is to ensure that at least our desired generic font family gets rendered on the result.

The common practice is to start with the font we want to render and always end with a generic family.

**Syntax:**

font family: font-name/s, generic family

Example: We can define the font family in the following way.

font family: Arial, sans-serif;

Here the browser will choose Arial first as it is specified first. If it is not installed on our device, it will set sans-serif as the generic fallback option.

**Note:** If a font name contains white space, it must be quoted with double quotes (" ").

**Types of Font Family in CSS**

There are two types of font families in CSS.

### Specific Font Family in CSS

As the name suggests, they are used only for a specific type of font. Specific font families in CSS have specific fonts with various styles within the one font family name.

These include **Arial, Times New Roman, Tahoma**, etc.

For example, **Arial** is a specific font within the **Sans-Serif** font family.

Take the example of the Arial font family. It consists of various fonts like Arial bold and Arial italic, which are just variations of the Arial font family. They differ in terms of their weights, so they considered font styles in CSS.

**Example:**

font family: Arial;

### Generic Font Family in CSS

The Generic font family in CSS is used for a general font, and almost all browsers support them. The generic font family is determined by the font family properties, such as cursive strokes or decorative strokes at the edge of the letters.

**There are five generic font families in CSS:**

**1. Serif-** Serif is a generic font family that includes fonts having small, winged tips or strokes at the end of each letter. Such strokes are known as Serifs. They are traditional typefaces. They are mostly used in textbooks, newspapers, etc.

**Example:** Times New Roman, Georgia etc.

The above fonts in the Serif font family have common properties of Serifs, and they are similar to each other.

Serif is a generic font family, while Times New Roman and Georgia are font families within the Serif family.

**Note:** Here, we are calling Times New Roman and Georgia font-families as they are specific font families and have fonts within their own families for different weights, i.e., font styles in CSS(bold, italic, etc.).

Like Times New Roman has Times New Roman bold and Times New Roman Italic. They differ in their font style.

**Appearance-**



**2. Sans-serif-** Sans-serif is a generic font family that includes fonts that are much cleaner and do not have those strokes at the end of each letter. It literally means "without Serif". They are commonly seen in digital spaces like articles etc.

**Example:** Arial, Helvetica, Verdana, Trebuchet MS, and Gill Sans.

The above fonts in the Sans-serif font family have common properties of having no strokes, and they are similar to each other.

Sans-serif is a generic font family, while Arial and Helvetica are font families within the Sans-serif family.

**Appearance-**



**3. Monospace-** They have letters of the same width. It renders a mechanical look.

**Example:** Courier New, Monaco, Lucida Console, Consolas, and Everson Mono.

The above fonts in the Monospace font family have the common properties of having letters with the same width, and they are similar to each other.

Monospace is a generic font family, while Courier New and Monaco are font families within the Monospace family.

**Appearance-**



**4. Cursive-** They are used to join the letters using connective strokes and give them a handwritten look.

**Example:** Comic Sans MS, Adobe Poetica, Sanvito, and Zapf-Chancery.

The above fonts in the Cursive font family have common properties of having connective strokes, and they are similar to each other.

Cursive is a generic font family, while Comic Sans MS and Adobe Poetica are fonts within the Cursive family.

**Appearance-**



**5. Fantasy-** They are decorative fonts that maintain the characteristics of non-cursive, traditional alphabet glyphs.

**Example:** Cottonwood, Critter, and Alpha Geometrique.

The above fonts in the Fantasy font family have common properties of having non-cursive characters, and they are similar to each other.

Fantasy is a font family, while Cottonwood and Critter are font families within the Fantasy family.

**Appearance-**



**Difference Between Serif and Sans-Serif Fonts**

****

In the above image, we can see that the Serif font has the strokes at the end of the letters, which are called serifs, while the Sans-serif font is cleaner and does not have the strokes at the end of the letters.

Serif is used widely in prints like magazines and newspapers but is not displayed well on screen. For this purpose, Sans-serif fonts are suitable to be used on computer screens.

**CSS Font Properties:**

CSS font properties allow the developers to manipulate how the font will look and feel on the webpage. The size, style, color, and much more can be changed by using properties. For e.g., if the developer wants to make a part of the text more impactful, he/she can BOLD the part.

The different CSS font properties are-

* CSS Font-color
* CSS font-size
* CSS font-style
* CSS font-variant
* CSS font-weight
* CSS font-family
* CSS font-stretch
* CSS line-height

**CSS Font-size:**

In CSS, the font-size property is used to set or tweak the size of the font. It can have several values that can be absolute (e.g.- xx-small, medium, xx-large.) or relative (larger, smaller, %) or length (numbered- 12px, 1em, etc.)

**Syntax:**

font-size: xx-small | x-large | xx-large | larger | smaller | 20% | 2em | 7px;

Any of the values can be used for the font-size property.

Now, we will talk about the different types of values that can be assigned to the font-size property.

**Absolute size:**

The absolute size is used to set the font size based on the predefined sizes. The default value of absolute size is medium. It can have values starting from xx-small to xx-large.

**Absolute size syntax:**

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

**Relative-size :**

Although the absolute-size is easy to use, it gives us less flexibility, and as a developer, to be more inclusive of all the users, we use relative-size. It sets the font relative to the parent element's font size. It can have two values- smaller and larger.

Now, suppose a parent element has a font size of medium, and then we make a child element and give its font-size: larger, then it will assign a value which is larger than 'medium' size. Finally, the size of the child element will be "large".

**Relative-size Syntax:**

font-size: larger | smaller;

**Length:** We can also use numerical length values like 23,54, etc., with units such as px, cm, etc. It gives more control over the size of the text.

**font-size: 10px | 12cm | 1em;**

**em** is the measure of the default text size in browsers, 1em= 16px.

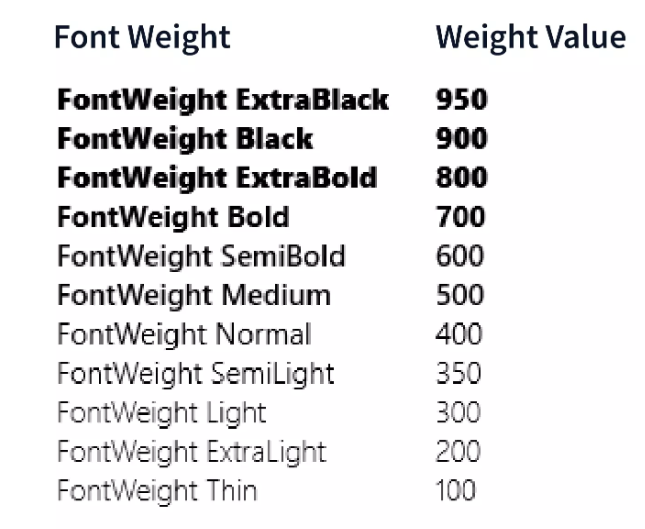
**CSS Font-Weight:**

The weight of the font refers to the thickness of the font, i.e., how thick or thin the font will be on the webpage. The font-weight property of CSS is used to set the weight/thickness/boldness of the font.

It is either dependent on the specified weights of the browser or the available font faces in a font family.

It can have absolute or relative, or numeric values like normal, bold, lighter, bolder, 100, 400. Though, We do not need to specify any unit while using the numeric values.

400 has the same weight as normal, and 700 have the same as bold.

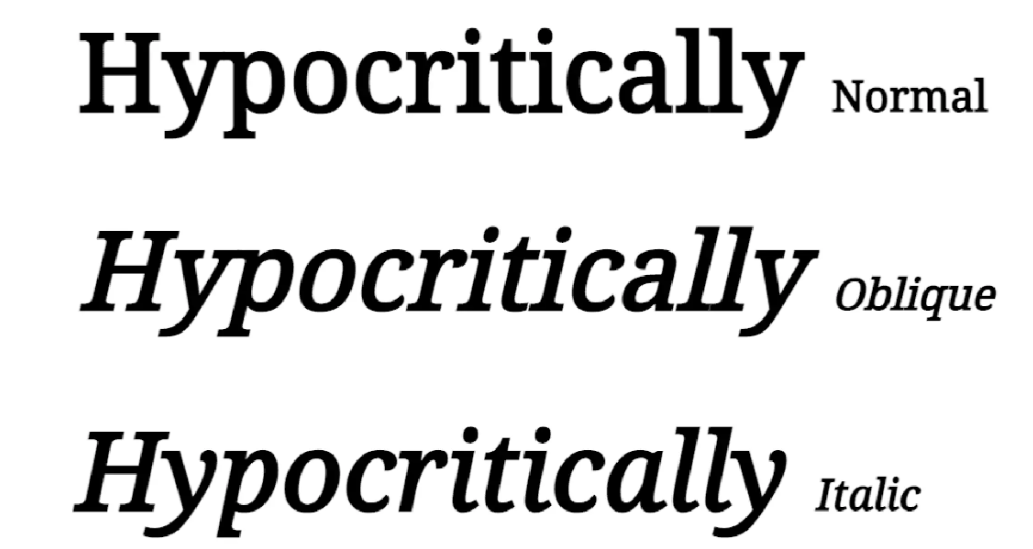


**CSS Font-Style:**

CSS font-style property is used to define the style of font for the text content of an element. Here, style refers to the variation in the typeface. It may be italic, oblique, or normal(default). Font-style property can be used to decorate and assign importance to a specific text.

**Italic vs Oblique:**

Italic forms are generally cursive in nature, while oblique faces are sloped versions of the regular face.



**Syntax:**

font-style: normal | italic | oblique;

**CSS Font-Variant:**

The font-variant property in CSS allows you to transform the specific line of text or paragraph into **small caps** (capital letters that will be small in size as compared to normal capital letters). It can also have values such as **normal** (by default), **initial**- This sets the property to the default value, and **inherit**- It takes the property's value of the parent element to the child element.

**Syntax:**

font-variant: normal | small-caps | initial | inherit;

Additional variants available in CSS3 are small-caps, all-small-caps, petite-caps, all-petite-caps, titling-caps, unicase, and titling-caps.

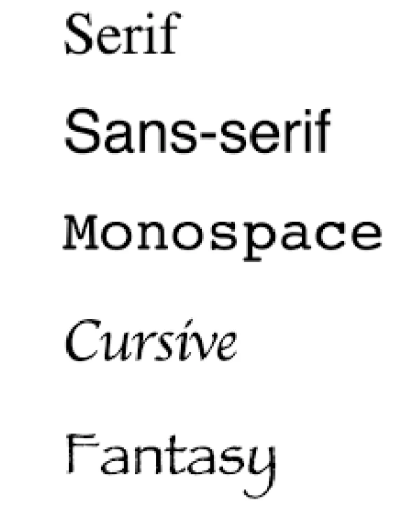
**CSS Font-Family:**

A font family is a set of fonts having similar typefaces and common designs. It is important to choose the right font family because fonts have many associated values. The right font (along with the right color and size) leaves a huge impact on readers and also shows your intention and emotion to put that font and make your identity. Font-family property can have values like family name (times, courier, arial, etc.) and generic-family (serif, sans-serif, cursive, fantasy, monospace)

In CSS, we use the font-family property to specify the family/ genre of the font on the webpage.

In CSS, there are five generic font families:

* **Serif** – All characters have a stroke attached at the endings.
* **Sans-serif** -These don’t have small strokes attached at endings.
* **Cursive** – similar to human cursive writing, the letters are partially or fully connected.
* **Monospace** - all the letters have the same fixed width.
* **Fantasy** – Used for decoration and playful text.



All the different font names belong to one of the generic font families.

The font-family property in CSS can take values as a prioritized list, i.e., It can hold various font names as a "fallback" system. The first mentioned font will have the highest priority, and if the browser doesn't support the first one, it will jump to the next mentioned font. These fonts should be separated by commas.

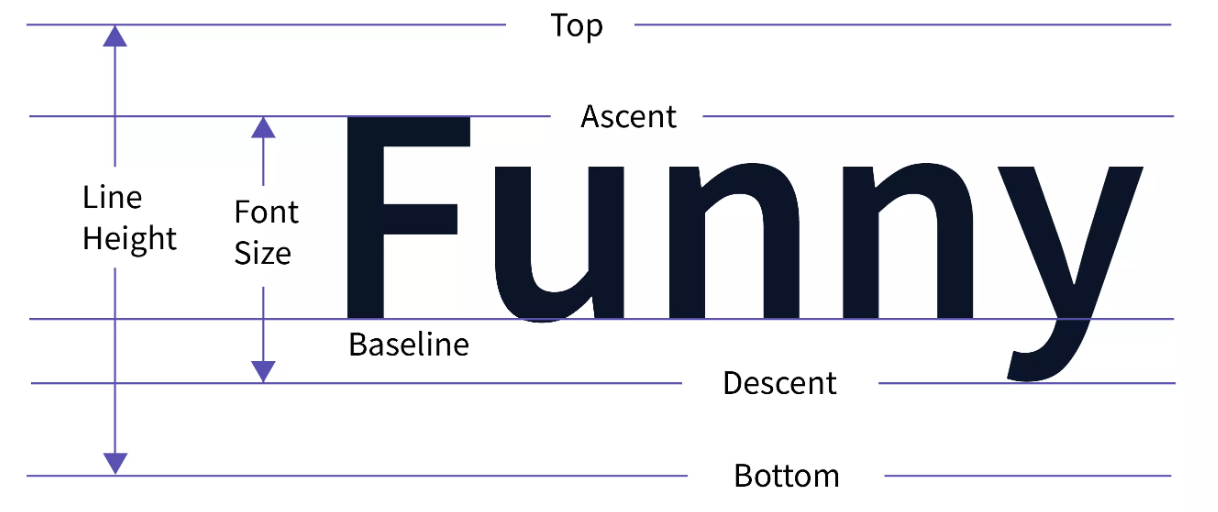


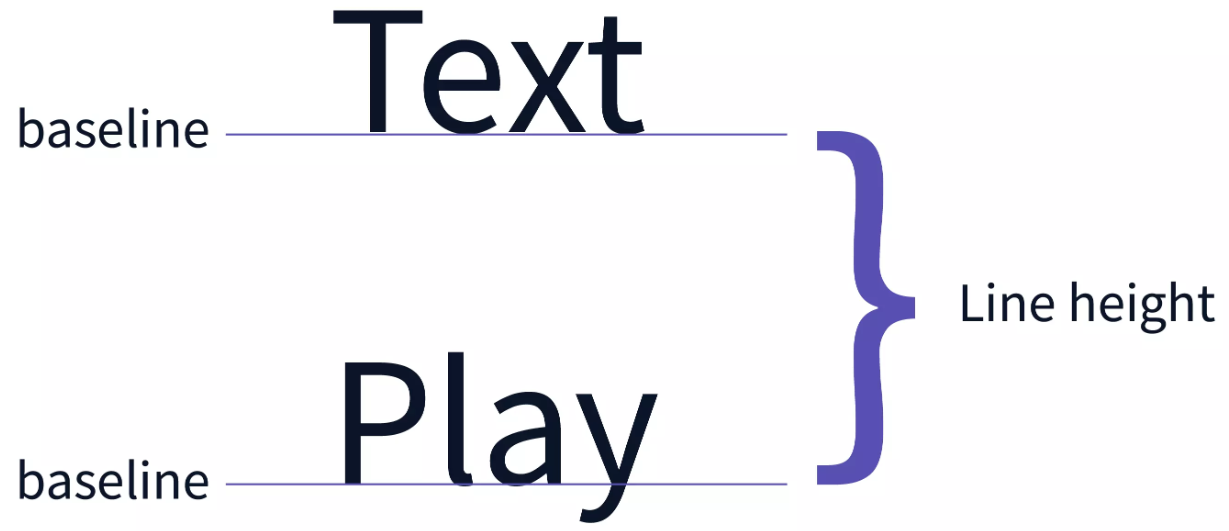
**Syntax:**

font-family: family-name|generic-family|initial;

**CSS Line-Height:**

Line height can be thought the distance between lines of text on a webpage. The line-height property in CSS allows us to define the height of the line box, i.e., the amount of space above and below the inline elements. It accepts the values such as normal, number, length, percentage, initial.





**Syntax:**

line-height: normal|number|length|percentage|initial;

**CSS Font (Shorthand) Property**

Instead of writing all the CSS Font-Properties one by one, we can use the font property, which is a shorthand property to define the font-style, font-variant, font-weight, font-size/line-height, and font-family. At least font-size and font-family should be provided for the shorthand to work. If one of the two is not provided, then we will not get the desired output as the default values of the font-size or the font-family will be used.

**Syntax:**

font: <font-style> <font-variant> <font-weight> <font-size/line-height> <font-family>;