# **CSS Fonts**

In this tutorial you will learn how to style fonts on a web page using CSS.

## **Styling Fonts with CSS**

Choosing the right font and style is very crucial for the readability of text on a page.

CSS provide several properties for styling the font of the text, including changing their face, controlling their size and boldness, managing variant, and so on.

The font properties are: font-family, font-style, font-weight, font-size, and font-variant.

Let's discuss each of these font properties one by one in more detail.

## **Font Family**

The font-family property is used to specify the font to be used to render the text.

This property can hold several comma-separated font names as a *fallback* system, so that if the first font is not available on the user's system, browser tries to use the second one, and so on.

Hence, list the font that you want first, then any fonts that might fill in for the first if it is not available. You should end the list with a generic font family which are five — serif, sans-serif, monospace, cursive and fantasy. A typical font family declaration might look like this:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=font-family)

body { font-family: Arial, Helvetica, sans-serif; }

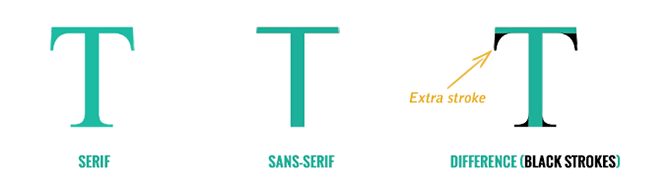
**Note:** If the name of a font family contains more than one word, it must be placed inside quotation marks, like "Times New Roman", "Courier New", "Segoe UI", etc.

The most common font families used in web design are *serif* and *sans-serif*, because they are more suitable for reading. While *monospace* fonts are commonly used to display code, because in this typeface each letter takes up the same space which looks like typewritten text.

The *cursive* fonts look like cursive writing or handwriting. The *fantasy* font represents artistic font, but they're not used widely because of poor availability across the operating systems.

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

Serif fonts have small line or stroke at the extremities of characters, whereas sans-serif fonts are straighter and do not have these small strokes. See the following illustration.



To learn about commonly used font combinations, please check out the reference on [web safe fonts](https://www.tutorialrepublic.com/css-reference/css-web-safe-fonts.php).

## **Font Style**

The font-style property is used to set the font face style for the text content of an element.

The font style can be normal, italic or oblique. The default value is normal.

Let's try out the following example to understand how it basically works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=font-style)

p.normal { font-style: normal; } p.italic { font-style: italic; } p.oblique { font-style: oblique; }

**Note:** At first glance both oblique and italic font styles appear the same thing, but there is a difference. The italic style uses an *italic version* of the font while oblique style on the other hand is simply a slanted or sloped version of the normal font.

## **Font Size**

The font-size property is used to set the size of font for the text content of an element.

There are several ways to specify the font size values e.g. with keywords, percentage, pixels, ems, etc.

### **Setting Font Size with Pixels**

Setting the font size in pixel values (e.g. 14px, 16px, etc.) is a good choice when you need the pixel accuracy. Pixel is an absolute unit of measurement which specifies a fixed length.

Let's try out the following example to understand how it basically works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-pixels)

h1 { font-size: 24px; } p { font-size: 14px; }

Defining the font sizes in pixel is not considered very accessible, because the user cannot change the font size from the browser settings. For instance, users with limited or low vision may wish to set the font size much larger than the size specified by you.

Therefore, you should avoid using the pixels values and use the values that are relative to the user's default font size instead if you want to create an inclusive design.

**Tip:** The text can also be resized in all browsers using the *zoom feature*. However, this feature resizes the entire page, not just the text. The [W3C](http://www.w3.org/Style/CSS/Overview.en.html) recommends using the em or percentage (%) values in order to create more robust and scalable layouts.

### **Setting Font Size with EM**

The em unit refers to the font size of the parent element. When defining the font-size property, 1em is equal to the size of the font that applies to the *parent of the element*.

So, if you set a font-size of 20px on the body element, then 1em = 20px and 2em = 40px.

However, if you haven't set the font size anywhere on the page, then it is the browser default, which is normally 16px. Therefore, by default 1em = 16px, and 2em = 32px.

Let's take a look at the following example to understand how it basically works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-em)

h1 { font-size: 2em; /\* 32px/16px=2em \*/ } p { font-size: 0.875em; /\* 14px/16px=0.875em \*/ }

### **Using the Combination of Percentage and EM**

As you've observed in the above example the calculation of em values doesn't look straightforward. To simplify this, a popular technique is to set the font-size for the body element to 62.5% (that is 62.5% of the default 16px), which equates to 10px, or 0.625em.

Now you can set the font-size for any elements using em units, with an easy-to-remember conversion, by dividing the px value by 10. This way 10px = 1em, 12px = 1.2em, 14px = 1.4em, 16px = 1.6em, and so on. Let's take a look at the following example:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-percentage-and-em)

body { font-size: 62.5%; /\* font-size 1em = 10px \*/ } p { font-size: 1.4em; /\* 1.4em = 14px \*/ } p span { font-size: 2em; /\* 2em = 28px \*/ }

### **Setting Font Size with Root EM**

To make things even more simpler CSS3 has introduced rem unit (short for "root em") which is always relative to the font-size of the root element (html), regardless of where the element lies in the document (unlike em which is relative to parent element's font size).

This means that 1rem is equivalent to the font size of the html element, which is 16px by default in most browsers. Let's try out an example to understand how it actually works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-rem)

html { font-size: 62.5%; /\* font-size 1em = 10px \*/ } p { font-size: 1.4rem; /\* 1.4rem = 14px \*/ } p span { font-size: 2rem; /\* 2rem = 20px (not 28px) \*/ }

### **Setting Font Size with Keywords**

CSS provide several keywords that you can use to define font sizes.

An absolute font size can be specified using one of the following keywords: xx-small, x-small, small, medium, large, x-large, xx-large. Whereas, a relative font size can be specified using the keywords: smaller or larger. Let's try out an example and see how it works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-keywords)

body { font-size: large; } h1 { font-size: larger; } p { font-size: smaller; }

**Note:** The keyword medium is equivalent to the browsers default font-size, which is normally 16px. Likewise, xx-small is the equivalent of 9 pixels, x-small is 10 pixels, small is 13 pixels, large is 18 pixels, x-large is 24 pixels, and xx-large is 32 pixels.

**Tip:** By setting a font size on the body element, you can set the relative font sizing everywhere else on the page, giving you the ability to easily scale the font size up or down accordingly.

### **Setting Font Size with Viewport Units**

The font sizes can be specified using viewport units such as vw or vh.

Viewport units refer to a percentage of the browser's viewport dimensions, where 1vw = 1% of viewport width, and 1vh = 1% of viewport height. Hence, if the viewport is 1600px wide, 1vw is 16px.

Try out the following example by resizing the browser window and see how it works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=define-font-size-using-viewport-units)

body { font-size: 1vw; } h1 { font-size: 3vw; }

However, there is a problem with the viewport units. On small screens fonts become so small that they are hardly readable. To prevent this you can utilize CSS calc() function, like this:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=fixing-font-size-issue-with-viewport-units)

html { font-size: calc(1em + 1vw); } h1 { font-size: 3rem; }

In this example, even if the viewport width becomes 0, the font-size would be at least 1em or 16px.

You further utilize [CSS media queries](https://www.tutorialrepublic.com/css-tutorial/css3-media-queries.php) to create better responsive and fluid typography.

## **Font Weight**

The font-weight property specifies the weight or boldness of the font.

This property can take one of the following values: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 and [inherit](https://www.tutorialrepublic.com/definitions.php#inherit).

* The numeric values 100-900 specify the font weights, where each number represents a weight greater than its predecessor. 400 is same as normal & 700 is same as bold.
* The bolder and lighter values are relative to the inherited font weight, while the other values such as normal and bold are absolute font weights.

Let's try out an example to understand how this property basically works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=font-weight)

p { font-weight: bold; }

**Note:** Most of the fonts are only available in a limited number of weights; often they are available only in *normal* and *bold*. In case, if a font is not available in the specified weight, an alternate will be chosen that is the closest available weight.

## **Font Variant**

The font-variant property allows the text to be displayed in a special small-caps variation.

Small-caps or *small capital letters* are slightly different to normal capital letters, in which lowercase letters appear as smaller versions of the corresponding uppercase letters.

Try out the following example to understand how this property actually works:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=css&file=font-variant)

p { font-variant: small-caps; }

The value normal removes small caps from the text which is already formatted that way.