

Common CSS properties

There are hundreds of CSS properties for you to use. The complete list is available from <https://www.w3.org/Style/CSS/all-properties.en.html> (or also at <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>).

Below we've gathered a more manageable list of the most useful and common CSS properties: `font-size`, `line-height`, `text-align`, `text-decoration`, `font-weight`, `font-style` and `font-family`.

FONT-SIZE

`font-size` can be used to size the text of a tag. The value for the `font-size` has two parts: a number and a unit. Some of the most common units are: `px`, `em`, `%`, `vh` . For example:

```
p { font-size: 18px; }
q { font-size: .8em; }
blockquote { font-size: 10vh; }
```

these units are discussed below.

Additionally, `font-size` supports a more readable set of values that many authors prefer: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large` and relative sizing (relative to the text of the parent): `larger`, `smaller`

```
p { font-size: medium; }
q { font-size: small; }
blockquote { font-size: larger; }
```

LINE-HEIGHT

Whereas `font-size` may drive the size of the text itself, the `line-height` property drives the height of the space it is drawn into. A large `line-height` will give the text more spacing. A small `line-height` will smash the text lines together.

For example, all of the Middlemarch text below has `font-size:16px;` But on the left we see `line-height:8px;` and on the right, `line-height:30px;`

<code>line-height: 8px;</code>	<code>line-height:30px;</code>
Miss Brooke had that kind of beauty which seems to be thrown into relief by poor dress.	Miss Brooke had that kind of beauty which seems to be thrown into relief by poor dress.

`line-height` supports the same *units* that `font-size` does (px, em, %, vh, etc)

TEXT-ALIGN

Anyone familiar with a text editor will be familiar with this property. Can be used to align the text `left`, `center` or `right`. There are additional possible values like `justify` and `justify-all`. Usually defaults to left.

Note that text-align may **not** work as expected if applied to elements that are the same width as their text, or whose width is determined by the text within them (ie inline elements). The tags ``, `<a>`, `<i>`, ``, `<q>` and others are considered "inline" because they do not receive their own new line when used. And `text-align` is often not useful on these tags.

But it is useful on block level text tags, such as `<p>`, ``, ``, ``, `<div>`, and `<blockquote>`

```
p { text-align: left; }
blockquote { text-align: right; }
```

Bear in mind, also, that you should only use text-align when the alignment really needs to be changed, since it can cause additional work to reverse all the values when translating into languages that use Arabic, Hebrew, Thaana, etc, scripts. (The default alignment for those languages is right.) The new values start and end are currently being implemented in browsers, and those will be a much better choice than left and right once Internet Explorer supports them.

left	center
------	--------

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair.	It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair.
right	justify
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair.	It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair.

TEXT-DECORATION (UNDERLINE)

How do I underline text? is a common question. In CSS this is done via the text-decoration property. The values for this are: underline,overline, line-through, and none; They can combined.

```
p { text-decoration: underline; }
a { text-decoration: none; } /* hyperlinks are underlined by default, but that can be removed */
span { text-decoration: overline; }
span { text-decoration: underline overline; } /* apply two with just a space between the values */
span { text-decoration: underline overline line-through; } /* everything */
```

underline	overline	line-through	underline overline line-through
<u>Middlemarch</u>	<u>Middlemarch</u>	Middlemarch	<u><u>Middlemarch</u></u>

Note: there are other properties that can help customize the text decoration, such as `text-decoration-color` and `text-decoration-skip`, but as of this writing they are not well supported (Firefox only).

FONT-WEIGHT (BOLD)

Earlier we saw that the `` and `` tags would make text bold-faced. But, semantically speaking, that is a mere side-effect of the tag. Any tag can make the text bolder (or less bold) via the `font-weight` CSS property. Common values are `normal` and `bold`. But text can also be made bolder (or less bold) than its parent with the values `bolder` and `lighter`. Lastly, the `font-weight` can be set explicitly as a numeric value. The choices are: `100`, `200`, `300`, `400`, `500`, `600`, `700`, `800` and `900`. `normal` maps to `400` and `bold` to `700`. But the different numeric choices will only work for fonts that support a full range of font-weights. Many times the numeric weights will simply be mapped back to bold or normal.

```
p { font-weight: bold; }
blockquote { font-weight: 900; }
```

normal	bold	200	500	700	900
A Tale of Two Cities	A Tale of Two Cities	A Tale of Two Cities	A Tale of Two Cities	A Tale of Two Cities	A Tale of Two Cities

FONT-STYLE (ITALIC)

Earlier we saw that the `<i>` and `` tags could make text italicized. But, just as we saw when discussing `font-weight`, this can be changed with CSS, and any tag can make its text italic or oblique with the `font-style` property. The choices of values for this property are `normal` and `italic`.

<code>font-style: normal;</code>	<code>font-style: italic;</code>
Many years later, as he faced the firing squad, Colonel Aureliano	<i>Many years later, as he faced the firing squad, Colonel Aureliano</i>

Buendía was to remember that distant afternoon when his father took him to discover ice.

Buendía was to remember that distant afternoon when his father took him to discover ice.

FONT-FAMILY

Want to set the font for an item on the page? The `font-family` is the correct property for the task, but there are caveats:

- the various browsers only guarantee a few standard choices: `serif`, `sans-serif`, `monospace`, `cursive`, and `fantasy`.
- any other choice must be already installed on the users machine
- or you may use a "Web font", but your choices, while plentiful, may not match the choices you are used to.
- your favorite font on your machine is probably encumbered by licensing limitations and is not available. You can certainly specify it to be used, but if the end user doesn't have it themselves, they won't see it. And you can't "give" it to them. Again, "Web fonts" are the alternative here.

To help ameliorate these limitations, the font-family property accepts a *list* of possible font choices. The browser will start with trying the first font listed, and if not available (or not having a needed glyph) it will then proceed to the next font in the list, and so on. Here is a typical font-family declaration:

```
p { font-family: "Helvetica", "Verdana", "Arial", sans-serif; }
```

The rule above says to first try the font named "Helvetica". If it isn't available, try "Verdana", failing that "Arial", and lastly fall back to the built in `sans-serif` browser font.

- each of the named font families is separated by a comma (,)
- if the font family name contains any spaces (or certain other characters) it **must** be surrounded by quotes. Font names tend to be complex, and the exact rules for when quotes are required are arcane, so the simplest and best practice is to **always surround the font family name in quotes**, excepting the five built-ins (`serif`, `sans-serif`, etc)

Web fonts are outside the scope of this course. Google provides a nice selection of license free Web fonts. Type "*google Web font tutorial*" into a search engine to learn more.