

CSS

CSS → **Cascading Style Sheets**

Font Family

```
P { font-family: "New Century Schoolbook", Times, serif; }
```

The `font-family` CSS property specifies a prioritized list of one or more font family names and/or generic family names for the selected element.

Font weight

```
/* Keyword values */
font-weight: normal;
font-weight: bold;

/* Keyword values relative to the parent */
font-weight: lighter;
font-weight: bolder;

/* Numeric keyword values */
font-weight: 100;
font-weight: 200;
font-weight: 300;
font-weight: 400; /* normal */
font-weight: 500;
font-weight: 600;
font-weight: 700; /* bold */
font-weight: 800;
font-weight: 900;

/* Global values */
font-weight: inherit;
font-weight: initial;
font-weight: revert;
font-weight: revert-layer;
font-weight: unset;
```

The `font-weight` CSS property sets the weight (or boldness) of the font. The weights available depend on the `font-family` that is currently set.

Font Size

```
/* <absolute-size> values */
font-size: xx-small;
font-size: x-small;
font-size: small;
font-size: medium;
font-size: large;
font-size: x-large;
font-size: xx-large;
font-size: xxx-large;

/* <relative-size> values */
font-size: smaller;
font-size: larger;

/* <length> values */
font-size: 12px;
font-size: 0.8em;

/* <percentage> values */
font-size: 80%;

/* math value */
font-size: math;

/* Global values */
font-size: inherit;
font-size: initial;
font-size: revert;
font-size: revert-layer;
font-size: unset;
```

The `font-size` CSS property sets the size of the font. Changing the font size also updates the sizes of the font size-relative `<length>` units, such as `em`, `ex`, and so forth.

Color

Changes the color of text.

```
/* Keyword values */
color: currentcolor;

/* <named-color> values */
color: red;
color: orange;
color: tan;
color: rebeccapurple;
```

```

/* <hex-color> values */
color: #090;
color: #009900;
color: #090a;
color: #009900aa;

/* <rgb()> values */
color: rgb(34, 12, 64, 0.6);
color: rgba(34, 12, 64, 0.6);
color: rgb(34 12 64 / 0.6);
color: rgba(34 12 64 / 0.3);
color: rgb(34 12 64 / 60%);
color: rgba(34.6 12 64 / 30%);

/* <hsl()> values */
color: hsl(30, 100%, 50%, 0.6);
color: hsla(30, 100%, 50%, 0.6);
color: hsl(30 100% 50% / 0.6);
color: hsla(30 100% 50% / 0.6);
color: hsl(30 100% 50% / 60%);
color: hsla(30.2 100% 50% / 60%);

/* <hwb()> values */
color: hwb(90 10% 10%);
color: hwb(90 10% 10% / 0.5);
color: hwb(90deg 10% 10%);
color: hwb(1.5708rad 60% 0%);
color: hwb(0.25turn 0% 40% / 50%);

/* Global values */
color: inherit;
color: initial;
color: revert;
color: revert-layer;
color: unset;

```

Background-Color

The `background-color` CSS property sets the background color of an element.

```

/* Keyword values */
background-color: red;
background-color: indigo;

/* Hexadecimal value */
background-color: #bbff00; /* Fully opaque */
background-color: #bf0; /* Fully opaque shorthand */
background-color: #11ffee00; /* Fully transparent */
background-color: #1fe0; /* Fully transparent shorthand */
background-color: #11ffeeff; /* Fully opaque */
background-color: #1fef; /* Fully opaque shorthand */

```

```

/* RGB value */
background-color: rgb(255, 255, 128); /* Fully opaque */
background-color: rgba(117, 190, 218, 0.5); /* 50% transparent */

/* HSL value */
background-color: hsl(50, 33%, 25%); /* Fully opaque */
background-color: hsla(
    50,
    33%,
    25%,
    0.75
); /* 75% opaque, i.e. 25% transparent */

/* Special keyword values */
background-color: currentcolor;
background-color: transparent;

/* Global values */
background-color: inherit;
background-color: initial;
background-color: revert;
background-color: revert-layer;
background-color: unset;

```

Background-Image

The `background-image` CSS property sets one or more background images on an element.

```
background-image: url("../media/examples/lizard.png");
```

Background-Repeat

The `background-repeat` CSS property sets how background images are repeated

```

/* Keyword values */
background-repeat: repeat-x;
background-repeat: repeat-y;
background-repeat: repeat;
background-repeat: space;
background-repeat: round;
background-repeat: no-repeat;

/* Two-value syntax: horizontal | vertical */
background-repeat: repeat space;
background-repeat: repeat repeat;
background-repeat: round space;

```

```
background-repeat: no-repeat round;

/* Global values */
background-repeat: inherit;
background-repeat: initial;
background-repeat: revert;
background-repeat: revert-layer;
background-repeat: unset;
```

Background

The `background` shorthand CSS property sets all background style properties at once, such as color, image, origin and size, or repeat method.

```
/* Using a <background-color> */
background: green;

/* Using a <bg-image> and <repeat-style> */
background: url("test.jpg") repeat-y;

/* Using a <box> and <background-color> */
background: border-box red;

/* A single image, centered and scaled */
background: no-repeat center/80% url("../img/image.png");

/* Global values */
background: inherit;
background: initial;
background: revert;
background: revert-layer;
background: unset;
```

Text-Decoration

The `text-decoration` shorthand CSS property sets the appearance of decorative lines on text. It is a shorthand for `text-decoration-line`, `text-decoration-color`, `text-decoration-style` property.

```
text-decoration: underline;
text-decoration: overline red;
text-decoration: none;
/* Global values */
text-decoration: inherit;
text-decoration: initial;
text-decoration: revert;
```

```
text-decoration: revert-layer;  
text-decoration: unset;
```

Text-Align

The `text-align` [CSS](#) property sets the horizontal alignment of the inline-level content inside a block element or table-cell box.

```
/* Keyword values */  
text-align: start;  
text-align: end;  
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;  
text-align: justify-all;  
text-align: match-parent;  
  
/* Character-based alignment in a table column */  
text-align: ".";  
text-align: "." center;  
  
/* Block alignment values (Non-standard syntax) */  
text-align: -moz-center;  
text-align: -webkit-center;  
  
/* Global values */  
text-align: inherit;  
text-align: initial;  
text-align: revert;  
text-align: revert-layer;  
text-align: unset;
```

Line-Height

The `line-height` [CSS](#) property sets the height of a line box.

```
/* Keyword value */  
line-height: normal;  
  
/* Unitless values: use this number multiplied  
by the element's font size */  
line-height: 3.5;  
  
/* <length> values */  
line-height: 3em;  
  
/* <percentage> values */
```

```
line-height: 34%;

/* Global values */
line-height: inherit;
line-height: initial;
line-height: revert;
line-height: revert-layer;
line-height: unset;
```

Margin

```
/* Apply to all four sides */
margin: 1em;
margin: -3px;

/* top and bottom | left and right */
margin: 5% auto;

/* top | left and right | bottom */
margin: 1em auto 2em;

/* top | right | bottom | left */
margin: 2px 1em 0 auto;

/* Global values */
margin: inherit;
margin: initial;
margin: revert;
margin: revert-layer;
margin: unset;
```

Sets the margins of an element by specifying top, bottom, left and right margins -- all either specifying length or percentage.

Padding

space between the border and content of selector.

```
/* Apply to all four sides */
padding: 1em;

/* top and bottom | left and right */
padding: 5% 10%;

/* top | left and right | bottom */
padding: 1em 2em 2em;
```

```
/* top | right | bottom | left */
padding: 5px 1em 0 2em;

/* Global values */
padding: inherit;
padding: initial;
padding: revert;
padding: revert-layer;
padding: unset;
```

Border

Sets the width, style, and color of an element's border.

```
/* style */
border: solid;

/* width | style */
border: 2px dotted;

/* style | color */
border: outset #f33;

/* width | style | color */
border: medium dashed green;

/* Global values */
border: inherit;
border: initial;
border: revert;
border: revert-layer;
border: unset;
```

Width

Each block-level or replaced element can be given a width, specified as a length, a percentage, or as auto.

```
/* <length> values */
width: 300px;
width: 25em;

/* <percentage> value */
width: 75%;

/* Keyword values */
width: max-content;
width: min-content;
```



```
width: fit-content(20em);
width: auto;

/* Global values */
width: inherit;
width: initial;
width: revert;
width: revert-layer;
width: unset;
```

height

The **height** CSS property specifies the height of an element.

```
/* <length> values */
height: 120px;
height: 10em;
height: 100vh;

/* <percentage> value */
height: 75%;

/* Keyword values */
height: max-content;
height: min-content;
height: fit-content(20em);
height: auto;

/* Global values */
height: inherit;
height: initial;
height: revert;
height: revert-layer;
height: unset;
```

Position

The **position** CSS property sets how an element is positioned in a document.

```
position: static;
position: relative;
position: absolute;
position: fixed;
position: sticky;

/* Global values */
position: inherit;
```

```
position: initial;  
position: revert;  
position: revert-layer;  
position: unset;
```

static

The element is positioned according to the normal flow of the document.

The `top`, `right`, `bottom`, `left`, and `z-index` properties have *no effect*. This is the default value.

relative

The element is positioned according to the normal flow of the document, and then offset *relative to itself* based on the values of `top`, `right`, `bottom`, and `left`. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were `static`.

absolute

The element is removed from the normal document flow, and no space is created for the element in the page layout
