# Common CSS Properties

Note: This is intended as quick reference and therefore does not contain all possible properties/values. For a more comprehensive list, visit <http://www.w3schoolscom/cssref> or the official CSS 3 specification at <http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>

## Text Formatting

**font-size** - sets the size of the font. Valid units include pt, px, em, and %. The base size is 1em=12pt=16px=100%  
*Values:* Any pt, px, em or % unit  
*Example:* p{ font-size: 1.1em; font-size: 110%; font-size: 18px; font-size: 13pt; }

**font-family** – the name of the font to use. The two base fonts are serif and sans-serif, but many other types of fonts are supported across browsers. You may specify a comma separated list and the browser will look through it until it finds one that it has. You should always end the list with either serif or sans-serif  
*Values*: Arial |Courier |Helvetica |serif | sans-serif  
*Example:* p{ font-family: Helvetica, courier, sans-serif; }

**font-weight** – The boldness of the font  
*Values:* normal | bold  
*Example:* p{ font-weight: bold; }

**color** – the text color. Can be specified by name for many colors, or you can use an RGB hexadecimal value. The value for each RGB color is two hexadecimal digits (#000000 is black, #ff0000 is pure red, #00ff00 is pure green, #0000ff is pure blue, and #ffffff is white)  
*Values:* Any hexadecimal value from #000000 to #ffffff, or predefined color  
*Example:* p{ color: #003768; color: #ffffff; color: white;}

**text-align** – the location to align the text within the containing block.   
*Values:* left |center | right  
*Example:* p{ text-align: right; }

**text-decoration** – Places a line over, under or through the text  
*Values:* none | overline | underline | line-through  
*Example:* p{ text-decoration: underline; }

**text-transform** – Allows the text to be cased a specific way despite how it was originally typed in  
*Values:* none | lowercase | uppercase | capitalize  
*Example:* p{ text-transform: uppercase; }

**cursor** – Changes the mouse cursor that is displayed when it is hovered over the element  
*Values:* auto | crosshair | default | e-resize | grab | help | move | n-resize | ne-resize | nw-resize | pointer | progress | s-resize | se-resize | sw-resize | text | w-resize | wait | not-allowed | no-drop

## Block Sizing

**height** – The height of the block (not including padding/margin/border)  
*Values:* Any unit

**width** – The width of the block (not including padding/margin/border)  
*Values:* Any unit

**min-height** – The minimum height of the block for use when height is based on percentage (not including padding/margin/border)  
*Values:* Any unit

**min-width** – The minimum width of the block for use when width is based on percentage (not including padding/margin/border)  
*Values:* Any unit

**max-height** – The maximum height of the block for use when height is based on percentage (not including padding/margin/border)  
*Values:* Any unit

**max-width** – The maximum width of the block for use when height is based on percentage (not including padding/margin/border)  
*Values:* Any unit

**padding** – Spacing between the inside wall of the block and its contents. Can be specified as a single unit for a uniform padding, or as units for each side in the order top, right, bottom, left  
*Values:* Any unit  
*Example:* div{ padding: 1em; } div{ padding: 2px 4px 3px 5px; }

**margin** – Spacing between the outside wall of the block and any other elements  
*Values:* Any em, px or % unit

**top** – For absolute or relatively positioned blocks, the distance from the top of the parent positioning container  
*Values:* Any unit

**left** – For absolute or relatively positioned blocks, the distance from the left of the parent positioning container  
*Values:* Any unit

**bottom** – For absolute or relatively positioned blocks, the distance from the bottom of the parent positioning container  
*Values:* Any unit

**right** – For absolute or relatively positioned blocks, the distance from the right of the parent positioning container  
*Values:* Any unit

**border** – style, color and size of the wall of the block  
*Values:* (solid | dashed ) (Any unit) (Any color value)  
*Example:* div { border: solid 2px #00ff00; }

**box-sizing** – Specifies the computation method for determining the final box size from the height/width/margin/border/padding   
*Values:* content-box | border-box

## Block Formatting

**background-color** – The background color of the block  
*Values:* Any color value. Can be specified by name, hexadecimal, or RGB/RGBA value

**background-image** – The image to use as the background  
*Values:* url(‘url\_to\_image\_here.jpg’)

**background-repeat** – Specifies which direction, if any, the image should repeat in. This can be used for an image that forms a border between elements  
*Values:* repeat | no-repeat | repeat-x | repeat-y

**background-position** – specifies where within the box you would like to place the image. Applies only when background-image is specified. You can specify both a horizontal and vertical position  
*Values:* left | right | top | bottom | <left unit value> <top unit value left unit value>

**float** – Allows a box to “float” to one side of the containing block or the other, letting other content flow around it  
*Values:* none | left | right

**clear** – specifies whether or not floating elements are allowed to float next to this element  
*Values:* none | left | right | both

**position** – specifies how the block will operate within the flow of the document  
*Values:* static | relative | absolute | fixed

Block TypeBe careful with this one. Before changing the display of an element you should ask yourself, “Am I really using the appropriate element here?”

**display** – How the block should be rendered  
*Values:* block | inline | inline-block | none | table | table-row | table-cell  
*Example:* span { display: block; }