



padding:

Just like when you eat too many donuts and expand, padding is like that, inside extending the elements background. It is a match to the background color.



border:

border encircles padding - it is usually a solid line, but can be dotted, or dashed (other options include double, groove, ridge, inset, outset.) You could think of it as the belt that surrounds Homer's donut gut.

You write it like so:

`border-right: 1px solid #fff;`

note: INLINE AND BLOCK borders are rendered differently

margin:

Margins are always transparent. They are a bit like rubber bumper cars. They aren't a continuation of the background, they act more like dividers, providing space

How to indicate sizes with the box model. Start in the upper left corner and proceed clockwise.

LONGHAND VERSION
`border-top: 4px;
border-right: 2px;
border-bottom: 1px;
border-left: 3px;`

SHORT HAND VERSION
`border: 4px 2px 1px 3px;`

FIRST VALUE FOR TOP AND BOTTOM
SECOND VALUE FOR LEFT AND RIGHT
`border: 6px 8px;`

TOP IS 2PX, RIGHT AND LEFT ARE 8PX, AND
BOTTOM IS 7PX
`border: 2px 8px 7px;`

between elements. Watch out for collapsing margins as happens when two horizontal margins are in contact with one another. If one margin is greater than the other, that will take precedent. Margin top and margin bottom will become one measurement.

* The star symbol stands for EVERYTHING on the web page. Most common usage of star is to reset a browser:

EXAMPLE

```
* { margin: 0; padding: 0; }
```

Id selector is the hash mark. When you assign this you can use the name only once. You make up the names you use for ids.

EXAMPLES

`#elmo`

● The class selector is a period. Use to target a group of elements.

EXAMPLE

`.muppets`

!important

Try not to use this. However, read about when it is good to use

<https://css-tricks.com/when-using-important-is-the-right-choice/>

width:

Used to set the width of the box. If you add padding, say 6px, that number will be *added* to the total width. So while your box may be specified as 30px wide, it will actually look like 42px wide!

Width properties have no effect on INLINE elements, unless you change the display property to block or inline-block.

min-width:, max-width:

Both of these will overrule width. The same holds true for max-height and min-width. Block-level elements will, by default, naturally expand to fit the available space, only the min-height value will actually have effect in many cases.

<https://www.impressivewebs.com/min-max-width-height-css/>

height:

Same deal as width in that padding does not get included. You can set width with px, % or ems and other units.

box-sizing: border-box;

This allows you to set the width *including* the padding.

FOR EXAMPLE

```
.element { box-sizing: border-box;  
width: 300px; padding: 20px; }
```

The width of the content area will be 300px - 40px = 260px.

CODE SNIPPET

```
html { box-sizing: border-box; }
```

```
*, *:before, *:after {  
box-sizing: inherit;  
}
```