Positioning, Sizing and Padding

CSS

Gayele Rizk

# Relative vs. Absolute Positioning

The CSS-Position property explains how elements are “positioned” on a web page. Usually, there exists 7 types of positioning: static, relative, absolute, fixed, sticky, initial, and inherit. However, the most Common Types are relative and absolute.

* Relative Positioning

Definition: When setting the position relative to a certain element excluding the necessity of adding position attributes such as the top, bottom, left, and right nothing will occur. However, when you add a certain action to the element such as the left position relative to the right of the element it will be done relatively to itself.

Note:

1. When the element gets shifted to a certain position no other element in the layout will be changed.
2. The relative positioning limits the scope of absolutely positioned child elements.

HTML

<div id="first\_element">First element</div>

<div id="second\_element">Second element</div>

CSS

#first\_element {

position: relative;

left: 30px;

top: 70px;

width: 500px;

background-color: #fafafa;

border: solid 3px #ff7347;

font-size: 24px;

text-align: center;

}

#second\_element {

position: relative;

width: 500px;

background-color: #fafafa;

border: solid 3px #ff7347;

font-size: 24px;

text-align: center;

}

* Absolute Positioning

Definition: Allows the positioning to occur exactly where you want to position your element. It is done relative to the first absolutely positioned parent element (When there is no positioned parent element, it will be positioned relative directly to the HTML element; page itself).

Note:

1. When using absolute positioning, do not overuse it for it does not contain proper maintenance for future purposes.

HTML

<div id=”parent”>

<div id=”child”></div>

</div>

CSS

#parent {

position: relative;

width: 500px;

height: 400px;

background-color: #fafafa;

border: solid 3px #9e70ba;

font-size: 24px;

text-align: center;

}

#child {

position: absolute;

right: 40px;

top: 100px;

width: 200px;

height: 200px;

background-color: #fafafa;

border: solid 3px #78e382;

font-size: 24px;

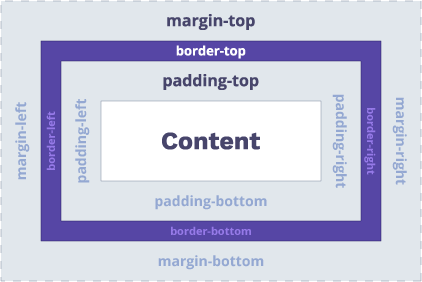
text-align: center;

}

## Box Sizing- More Precisely Margin vs. Padding

The Box-Sizing property can facilitate creating the CSS Layouts. The padding and Margin are part of a concept called “Box Model” (as shown below). We can see that the padding is the innermost layer of the box model whereas the margin adds space around elements in the box.

In addition, the margins can accept negative values whereas the padding cannot.

9

**Padding**

Padding portrays the idea of space on all four sides of a certain element; in other words, it is the area between the border and the content. Hence, it is considered within the elements.

With the use of the padding Shorthand, it allows the stating of the four properties of the padding with one mention.

Ex:

p {

padding: 30px;

}

The four properties include:

1. Padding-top: Sets the top padding of an element
2. Padding-right: Specifies the right padding of an element
3. Padding-bottom: Sets the bottom padding of an element
4. Padding-left: Specifies the left padding of an element
5. This property indicates the height of the padding area above an element.

Ex:

p {

padding-top: 20px;

}

1. This property shows the width of the padding area on the right of the element.

Ex:

p{

Padding-right: 20px;

}

1. This property denotes the height of the padding area below the element.

Ex:

p{

padding-bottom: 20px;

}

1. This property indicates the width of the padding area on the left side of the element.

Ex:

p{

padding-left: 40px;

}

To specify values for all the previous properties we can use 2 methods:

* Length to use the number and a unit: <length>
* Percentage to show the percentage value of an element: <percentage>
* Most times, when setting padding to elements with specific width, unexpected results will occur. The padding will be added to the width of the element. Padding won't be a part of the element. Developers see this as unwanted because the width of elements changes every time padding changes. This might also occur with height however height is hardly set.
* This issue can be fixed with box-sizing where it tells the element to keep its original width when increasing the padding. Also, it makes the available content space smaller.

Ex:

div {

width: 225px;

padding: 25px;

box-sizing: border-box;

}

Note:

* By leaving out values for specific properties in padding, you set those properties to their initial value of 0 (no space).
* Use CSS padding to define all four sides of elements and write shorter code.

Learn from the best web developers

**Margin**

Margin is what creates the gap around the content block, by “pushing” the content away. It's always outside of the border.

When to use margin:

* To create a space around the content block
* To make a space between two elements
* Center the content block *(using the same value for all sides)*.
* To create a space around the border.