Center Align Elements

To horizontally center a block element (like <div>), use margin: auto;

Setting the width of the element will prevent it from stretching out to the edges of its container.

The element will then take up the specified width, and the remaining space will be split equally between the two margins:

This div element is centered.

Example

.center {  
    margin: auto;  
    width: 50%;  
    border: 3px solid green;  
    padding: 10px;  
}

**Note:** Center aligning has no effect if the width property is not set (or set to 100%).

Center Align Text

To just center the text inside an element, use text-align: center;

This text is centered.

Example

.center {  
    text-align: center;  
    border: 3px solid green;  
}

Center an Image

To center an image, use margin: auto; and make it into a **block** element:



Example

img {  
    display: block;  
    margin: auto;  
    width: 40%;  
}

Left and Right Align - Using position

One method for aligning elements is to use position: absolute;:

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.

Example

.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}

**Note:** Absolute positioned elements are removed from the normal flow, and can overlap elements.

**Tip:** When aligning elements with position, always define margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is also a problem with IE8 and earlier, when using position. If a container element (in our case <div class="container">) has a specified width, and the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. So, always set the !DOCTYPE declaration when using position:

Example

body {  
    margin: 0;  
    padding: 0;  
}  
  
.container {  
    position: relative;  
    width: 100%;  
}  
  
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    background-color: #b0e0e6;  
}

Left and Right Align - Using float

Another method for aligning elements is to use the float property:

Example

.right {  
    float: right;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}

**Tip:** When aligning elements with float, always define margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is also a problem with IE8 and earlier, when using float. If the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. So, always set the !DOCTYPE declaration when using float:

Example

body {  
    margin: 0;  
    padding: 0;  
}  
  
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}

Center Vertically - Using padding

There are many ways to center an element vertically in CSS. A simple solution is to use top and bottom padding:

I am vertically centered.

Example

.center {  
    padding: 70px 0;  
    border: 3px solid green;  
}

To center both vertically and horizontally, use padding and text-align: center:

I am vertically and horizontally centered.

Example

.center {  
    padding: 70px 0;  
    border: 3px solid green;  
    text-align: center;  
}

Center Vertically - Using line-height

Another trick is to use the line-height property with a value that is equal to the height property.

I am vertically and horizontally centered.

Example

.center {  
    line-height: 200px;  
    height: 200px;  
    border: 3px solid green;  
    text-align: center;  
}  
  
/\* If the text has multiple lines, add the following: \*/  
.center p {  
    line-height: 1.5;  
    display: inline-block;  
    vertical-align: middle;  
}

Center Vertically - Using position & transform

If padding and line-height is not an option, a third solution is to use positioning and the transform property:

I am vertically and horizontally centered.

Example

.center {   
    height: 200px;  
    position: relative;  
    border: 3px solid green;   
}  
  
.center p {  
    margin: 0;  
    position: absolute;  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
}