# **CSS Margins**

Margin is used to create space around elements, outside of any defined borders.

By using margin we can give space around each side of an element (top, right, bottom, and left).

CSS has properties for specifying the margin for each side of an element:

* margin-top
* margin-right
* margin-bottom
* margin-left

### **Example**

p {  
  margin-top:50px;  
  margin-bottom: 100px;

margin-left:50px;  
  margin-right: 100px;

}

All the margin properties can have the following values:

* auto - the browser calculates the margin
* *length* - specifies a margin in px, pt, cm, etc.
* *%* - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element

### By using margin negative values we can give for HTML elements

### **Example**

p {  
  margin-top: -10px;

}

## Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The margin property is a shorthand property for the following individual margin properties:

* margin-top:2px
* margin-right:0px
* margin-bottom:5px
* margin-left:10px

So, here is how margin shorthand it works:

**margin: 2px 0px 5px 10px; (top right bottom left)**

left margin is 1If the margin property has three values:

**margin: 25px 50px 75px; (top right & left bottom)**

If the margin property has two values:

**margin: 25px 50px; (margin:top &bottom left&right)**

## The auto Value

You can set the margin property to auto to horizontally center the element within its container. When we give margin to any element the browser will calculates the width;

**Example**

div {  
  width: 300px;  
  margin: auto;  
  border: 1px solid red;  
}

## The inherit Value

## In CSS inherit means the child element inherit the properties from the parent element

This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

### **Example**

## div {   border: 1px solid red;   margin-left: 100px; } p.ex1 {   margin-left: inherit; }

## Margin Collapse

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

This does not happen on left and right margins! Only top and bottom margins!

Look at the following example:

### **Example**

h1 {  
  margin: 0 0 50px 0;  
}  
  
h2 {  
  margin: 20px 0 0 0;  
}

In the example above, the <h1> element has a bottom margin of 50px and the <h2> element has a top margin set to 20px.

Common sense would seem to suggest that the vertical margin between the <h1> and the <h2> would be a total of 70px (50px + 20px). But due to margin collapse, the actual margin ends up being 50px.

# **CSS Padding**

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

## Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

* padding-top
* padding-right
* padding-bottom
* padding-left

All the padding properties can have the following values:

* length - specifies a padding in px, pt, cm, etc.
* % - specifies a padding in % of the width of the containing element
* inherit - specifies that the padding should be inherited from the parent element

**Note:** Negative values are not allowed.

### **Example**

## div {   padding-top: 50px;   padding-right: 30px;   padding-bottom: 50px;   padding-left: 80px; }

## Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.

The padding property is a shorthand property for the following individual margin properties:

* padding-top:2px
* padding-right:0px
* padding-bottom:5px
* padding-left:10px

So, here is how padding shorthand it works:

**padding: 2px 0px 5px 10px; (top right bottom left)**

left margin is 1If the margin property has three values:

**padding: 25px 50px 75px; (top right & left bottom)**

If the padding property has two values:

**padding: 25px 50px; (padding:top &bottom left&right)**

## Box-sizing

The CSS box-sizing property allows us to include the padding and border in an element's total width and height.

By default, the width and height of an element is calculated like this:

width + padding + border = actual width of an element  
height + padding + border = actual height of an element

**Example**

div {  
  width: 300px;  
  padding: 25px;  
}

Use the box-sizing property to keep the width at 300px, no matter the amount of padding:

div {  
  width: 300px;  
  padding: 25px;  
  box-sizing: border-box;  
}

# **CSS Height and Width**

The CSS height and width properties are used to set the height and width of an element.

**Example**

div {

height: 50px;

width: 100%;

border: 1px solid #4CAF50;

}

In the above example the width of the div element is 100% and height of the element is 50px

## Height and Width Values

The height and width properties may have the following values:

* auto - This is default. The browser calculates the height and width
* length - Defines the height/width in px, cm etc.
* % - Defines the height/width in percent of the containing block
* initial - Sets the height/width to its default value
* inherit - The height/width will be inherited from its parent value

## Max-width

The max-width property is used to set the maximum width of an element.

### **Example**

This <div> element has a height of 100 pixels and a max-width of 500 pixels:

div {  
  max-width: 500px;  
  height: 100px;  
 }

## Max-height

The max-height property is used to set the maximum height of an element.

### **Example**

This <div> element has a max-height of 100 pixels and a width of 100%

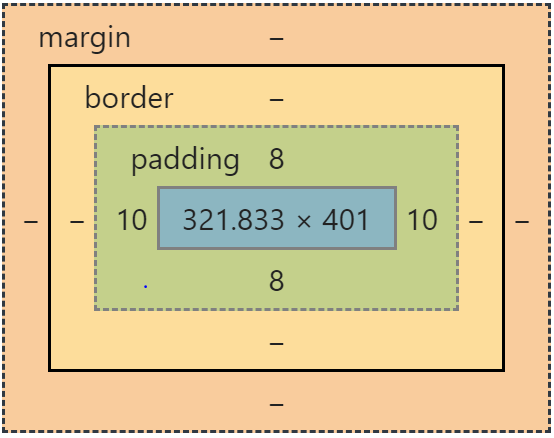
div {  
  max-height: 500px;  
  width: 100%;  
 }

## The CSS Box Model

All HTML elements can be considered as boxes. It consists of: margins, borders, padding, and the actual content. In CSS, the term "box model" is used when talking about design and layout.

The image below illustrates the box model:

The CSS box model is essentially a box that wraps around every HTML element.



* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent

## Outline

## An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".

## Example

## p {outline: #4CAF50 solid 10px;}

CSS has the following outline properties:

* outline-style
* outline-color
* outline-width
* outline-offset
* outline

## Outline Style

The outline-style property specifies the style of the outline, and can have one of the following values:

* dotted - Defines a dotted outline
* dashed - Defines a dashed outline
* solid - Defines a solid outline
* double - Defines a double outline
* groove - Defines a 3D grooved outline
* ridge - Defines a 3D ridged outline
* inset - Defines a 3D inset outline
* outset - Defines a 3D outset outline
* none - Defines no outline
* hidden - Defines a hidden outline

## Outline Width

The outline-width property specifies the width of the outline, and can have one of the following values

**Example:**

p {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: 4px;  
}

## Outline Color

## Example:

## p {   border: 2px solid black;   outline-style: solid;   outline-color: red; }

## Note:

## We can use hex values, hsl values, rgb values.

## CSS Outline - Shorthand property

The outline property is a shorthand property for setting the following individual outline properties:

* outline-width
* outline-style (required)
* outline-color

### **Example**

## p {outline: 5px solid yellow;}

## CSS Outline Offset

The outline-offset property adds space between an outline and the edge/border of an element. The space between an element and its outline is transparent.

### **Example**

## .myclass {   margin: 30px;   border: 1px solid black;   outline: 1px solid red;   outline-offset: 15px; }

# **CSS Text Properties**

1. Color
2. Text-align
3. Direction
4. Vertical-align
5. Text-decoration
6. Text-transform
7. Text-indent
8. Text-shadow

**1.Color:** The color property is used to set the color of the text. We can give values like

**Example**

div{color:red;}

div{color:#ff0000}

div{color:rgb(255,0,0)}

**2.text-align:** The text-align property is used to set the horizontal alignment of the text. We can give values like text-align:center, text-align:right, text-align:leftW

**Example**

div{text-align:center;}

div{text-align:left }

div{text-align:right}

**3.Direction:** The direction and unicode-bidi properties can be used to change the text direction of an element:

**Example**

p {  
  direction: rtl;unicode-bidi: bidi-override;  
}

**4.vertical-align:** The vertical-align property sets the vertical alignment of an element.

**Example**

p {  
  vertical-align: top /bottom/center   
}

**4.text-decoration:** The text-decoration property is used to set or remove decorations from text.

**Example**

p {  
  text-decoration: overline/line-through/underline  
}

**5.text-transform:** The text-transform property is used to specify uppercase and lowercase letters in a text.

**Example**

p {  
  text-transform: uppercase/lowercase/capitalize  
}

**Note:** by using uppercase we can change all the letters to uppercase, by using lowercase we can change all the letters to lower case, by using capitalize we can capitalize the first letter of each word in a sentence.

**6.text-indent:** The text-indent property is used to specify the indentation of the first line of a text:

**Example**  
p {  
  text-indent: 50px;  
}

**7.letter-spacing:** The letter-spacing property is used to specify the space between the characters in a text.

**Example**  
p {letter-spacing:5px }

We can also use ‘-’ values for letter spacing

p {letter-spacing:-5px }

**8.line-height:** The line-height property is used to specify the space between lines:

**Example**  
p {line-height:25px }

**9.line-height:** The line-height property is used to specify the space between lines:

**Example**  
p {line-height:25px }

**10.word-spacing:** The word-spacing property is used to specify the space between the words in a text.

**Example**  
p {word-spacing:5px }

p {word-spacing:-5px }

**11.white-space:**The white-space property specifies how white-space inside an element is handled.

**Example**  
p {  
  white-space: nowrap;  
}

**12.text-shadow:**The text-shadow property is used to give the shadow to the text.

**Example**  
.para p {  
  text-shadow: 2px 2px;  
}

We can also give color to the shadow

.para p  {  
  text-shadow: 2px 2px #ff0000;  
}

.para p  {  
  text-shadow: 2px 2px 5px #ff0000;  
}