CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

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

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

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

Tip: Negative values are allowed.

Example:

```
<style>

div {

border: 1px solid black;

margin-top: 100px;

margin-bottom: 100px;

margin-right: 150px;

margin-left: 80px;

background-color: lightblue;

}

</style>
```

Margin - Shorthand Property

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

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

So, here is how it works:

If the `margin` property has **four values:**

- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

Example:

<style>

div {

border: 1px solid black;

margin: 25px 50px 75px 100px;

background-color: lightblue;

}

</style>

If the `margin` property has three values:

- **margin: 25px 50px 75px;**
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px

example:

<style>

div {

border: 1px solid black;

margin: 25px 50px 75px;

background-color: lightblue;

}

</style>

If the margin property has two values:

- **margin: 25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px

Example:

<style>

div {

border: 1px solid black;

margin: 25px 50px;

background-color: lightblue;

}

</style>

If the margin property has one value:

- **margin: 25px;**
  - all four margins are 25px

Example:

```
p {
  margin: 25px;
}
```

The auto Value

You can set the margin property to `auto` to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

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

The inherit Value

This example lets the left margin of the element be inherited from the parent element ( ):

<style>

div {

border: 1px solid red;

margin-left: 100px;

}

p.ex1 {

margin-left: inherit;

}

</style>

</head>

<body>

<h2>Use of the inherit value</h2>

Let the left margin be inherited from the parent element:

This paragraph has an inherited left margin (from the div element).

</body>

CSS Padding

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

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

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.

<style>

div {

border: 1px solid black;

background-color: lightblue;

padding-top: 50px;

padding-right: 30px;

```
padding-bottom: 50px;

padding-left: 80px;

}

</style>
```

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 padding properties:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

So, here is how it works:

If the `padding` property has four values:

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px

Example:

```
div {
  padding: 25px 50px 75px 100px;
}
```

If the `padding` property has three values:

- **padding: 25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px

```
div {
  padding: 25px 50px 75px;
}
```

If the `padding` property has two values:

- **padding: 25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px

```
div {
  padding: 25px 50px;
}
```

If the `padding` property has one value:

- **padding: 25px;**
  - all four paddings are 25px

```
div {
  padding: 25px;
}
```

Padding and Element Width

The CSS `width` property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

<html>

<head>

<style>

div.ex1 {

width: 300px;

background-color: yellow;

}

div.ex2 {

width: 300px;

```
padding: 25px;

background-color: lightblue;

}

</style>

</head>

<body>

<h2>Padding and element width</h2>


<div class="ex1">This div is 300px wide.</div>

<br>

<div class="ex2">The width of this div is 350px, even though it is defined as 300px in the CSS.</div>

</body>

</html>
```

CSS Height, Width and Max-width

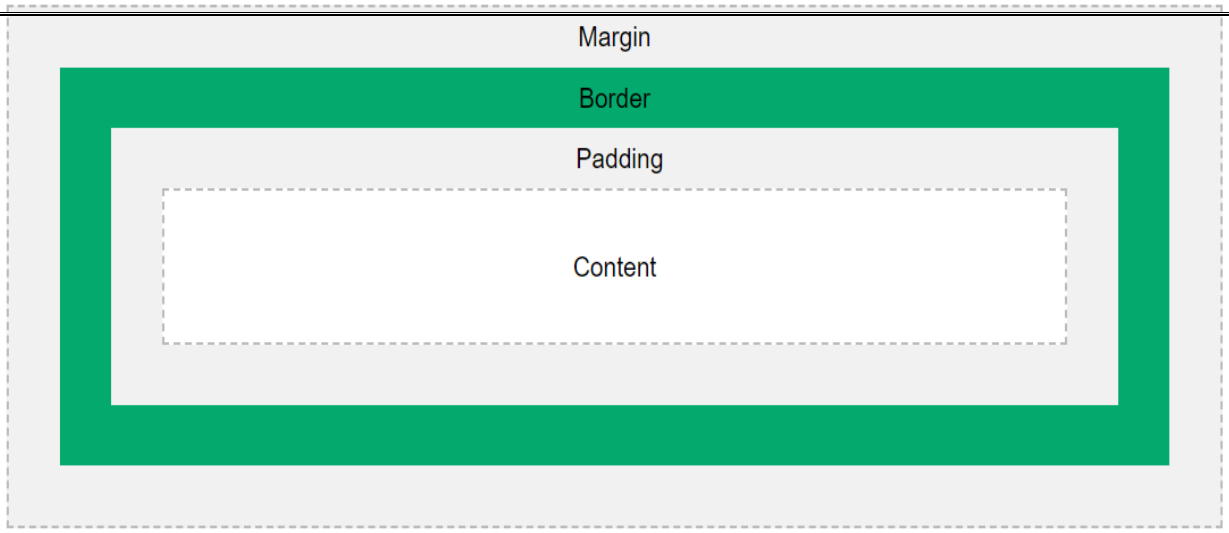
The CSS `height` and `width` properties are used to set the height and width of an element.

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

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- **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

The box model allows us to add a border around elements, and to define space between elements.

div {

background-color: lightgrey;

width: 300px;

border: 15px solid green;

padding: 50px;

margin: 20px;

}

</style>

CSS Outline

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

CSS has the following outline properties:

- `outline-style`
- `outline-color`
- `outline-width`
- `outline-offset`
- `outline`

CSS 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

CSS Text

Text Color

The `color` property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

CSS Text Alignment

Text Alignment and Text Direction

In this chapter you will learn about the following properties:

- `text-align`
- `text-align-last`

- direction
- unicode-bidi
- vertical-align

Text Alignment

The text-align property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

<style>

h1 {

text-align: center;

}

h2 {

text-align: left;

}

h3 {

text-align: right;

}

When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):

```
div {
  text-align: justify;
}
```

Text Align Last

The `text-align-last` property specifies how to align the last line of a text.

Text Decoration

- `text-decoration-line`
- `text-decoration-color`
- `text-decoration-style`
- `text-decoration-thickness`
- `text-decoration`

Add a Decoration Line to Text

The `text-decoration-line` property is used to add a decoration line to text.

Tip: You can combine more than one value, like overline and underline to display lines both over and under a text.

```
<style>

h1 {

text-decoration: overline;

}


h2 {

text-decoration: line-through;

}


h3 {
```

```
text-decoration: underline;
}

p.ex {
text-decoration: overline underline;
}
</style>
```

Specify a Color for the Decoration Line

The `text-decoration-color` property is used to set the color of the decoration line.

```
h1 {
  text-decoration-line: overline;
  text-decoration-color: red;
}

h2 {
  text-decoration-line: line-through;
  text-decoration-color: blue;
}

h3 {
  text-decoration-line: underline;
  text-decoration-color: green;
}

p {
  text-decoration-line: overline underline;
  text-decoration-color: purple;
}
```

Specify a Style for the Decoration Line

The `text-decoration-style` property is used to set the style of the decoration line.

```
h1 {
  text-decoration-line: underline;
  text-decoration-style: solid;
}
```

```
h2 {


 text-decoration-line: underline;
  text-decoration-style: double;
}

h3 {
  text-decoration-line: underline;
  text-decoration-style: dotted;
}

p.ex1 {
  text-decoration-line: underline;
  text-decoration-style: dashed;
}

p.ex2 {
  text-decoration-line: underline;
  text-decoration-style: wavy;
}

p.ex3 {
  text-decoration-line: underline;
  text-decoration-color: red;
  text-decoration-style: wavy;
}
```

Specify the Thickness for the Decoration Line

The `text-decoration-thickness` property is used to set the thickness of the decoration line.

```
h1 {
  text-decoration-line: underline;
  text-decoration-thickness: auto;
}

h2 {
  text-decoration-line: underline;
  text-decoration-thickness: 5px;
}

h3 {
  text-decoration-line: underline;
  text-decoration-thickness: 25%;
}
```

```
p {

 text-decoration-line: underline;
  text-decoration-color: red;
  text-decoration-style: double;
  text-decoration-thickness: 5px;
}
```

The Shorthand Property

The `text-decoration` property is a shorthand property for:

- `text-decoration-line` (required)
- `text-decoration-color` (optional)
- `text-decoration-style` (optional)
- `text-decoration-thickness` (optional)

- ```
  h1 {
    text-decoration: underline;
  }

  h2 {
    text-decoration: underline red;
  }

  h3 {
    text-decoration: underline red double;
  }

  p {
    text-decoration: underline red double 5px;
  }
  ```

Text Transformation

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

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

```
p.uppercase {
  text-transform: uppercase;
}
```
```

```
p.lowercase{

 text-transform: lowercase;
}

p.capitalize {
  text-transform: capitalize;
}
```

Text Spacing

- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

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

- ```
  p {
    text-indent: 50px;
  }
  ```

Letter Spacing

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

The following example demonstrates how to increase or decrease the space between characters:

```
h1 {
  letter-spacing: 5px;
}

h2 {
  letter-spacing: -2px;
}
```
```

Line Height

The `line-height` property is used to specify the space between lines:

```
p.small {
  line-height: 0.8;
}

p.big {
  line-height: 1.8;
}
```

Word Spacing

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

The following example demonstrates how to increase or decrease the space between words

```
p.one {
  word-spacing: 10px;
}

p.two {
  word-spacing: -2px;
}
```

White Space

The `white-space` property specifies how white-space inside an element is handled.

This example demonstrates how to disable text wrapping inside an element:

```
p {
  white-space: nowrap;
}
```

Text Shadow

The `text-shadow` property adds shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

CSS Font:

The CSS font properties allow you to change the font family, boldness, size, and the style of a

text.

Possible Values

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

Property	Value	Description
font-family	family-name generic-family	A prioritized list of font family names and/or generic family names for an element
font-size	xx-small x-small small medium large x-large xx-large smaller larger length, %	Sets the size of a font
font-style	normal italic oblique	Sets the style of the font
font-variant	normal small-caps	Displays text in a small-caps font or a normal font
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	Sets the weight of a font

CSS Border

The CSS border properties allow you to specify the style and color of an element's border. In HTML we use tables to create borders around a text, but with the CSS border properties we can create borders with nice effects, and it can be applied to any element.

Property	Value	Description
border-color	color	Sets the color of the four borders, can have from one to four colors
border	border-width border-style border-color	A shorthand property for setting all of the properties for the four borders in one declaration
border-bottom	border-bottom-width border- bottom - style border- bottom - color	A shorthand property for setting all of the properties for the bottom border in one declaration
border-bottom-color	border-color	Sets the color of the bottom border
border-bottom-style	border-style	Sets the style of the bottom border
border-bottom-width	thin medium thick length	Sets the width of the bottom border
border-left-color	border-color	Sets the color of the left border
border-left-style	border-style	Sets the style of the left border
border-left-width	thin medium thick length	Sets the width of the left border
border-right-color	border-color	Sets the color of the Right border
border-right-style	border-style	Sets the style of the Right border
border-right-width	thin medium thick length	Sets the width of the Right border
border-top-color	border-color	Sets the color of the top border
border-top-style	border-style	Sets the style of the top border
border-top-width	thin medium thick	Sets the width of the top border

```
p { border: medium double rgb(250,0,255) }
```

CSS List

The CSS list properties allow you to place the list-item marker, change between different listitem markers, or set an image as the list-item marker.

Property	Value	Description
list-style	list-style-type list-style-position list-style-image	A shorthand property for setting all of the properties for a list in one declaration
list-style-image	none url	Sets an image as the list-item marker
list-style-position	inside outside	Sets where the list-item marker is placed in the list
list-style-type	None, disc, circle square , decimal decimal-leading-zero ,lower-roman upper-roman ,lower-alpha upper-alpha	Sets the type of the list-item marker
padding-top	length %	Sets the top padding of an element

```
ul

{

list-style-image: url('arrow.gif')

}
```

CSS Table

The CSS table properties allow you to set the layout of a table.

Property	Value	Description
table-layout	auto fixed	Sets the algorithm used to display the table cells, rows, and columns
caption-side	top bottom left right	Sets the position of the table caption
border-spacing	length	Sets the distance that separates cell borders
empty-cells	show hide	Sets whether or not to show empty cells in a table

Example: - Set the layout of table. Task - Set the Space between table border.

```
<html>
<head>
<style
        type="t
        ext/css"
        >
        table.o
        ne
        {
                table-layout: automatic
        }
        table.two
        {
                table-layout: fixed
        }
</style>
</head>
<body>
<table class="one" border="1" width="100%">
<tr>
        <td width="20%">10000000000000000000000000000</td>
        <td width="40%">10000000</td>
        <td width="40%">100</td>

</tr>

</table>
<br />
<table class="two" border="1" width="100%">
<tr>
```

10000000000000000000000000000	10000000	100
1000000000(	10000000	100

```
</tr>
</table>
</body>
</html>
```

Output:

```
<td
width="20%">100
0000000000000000
000000000</td>
<td
width="40%">100
00000</td>
<td
width="40%">100
</td>
```

CSS Dimension

The CSS dimension properties allow you to control the height and width of an element. Italso allows you to increase the space between two lines.

Property	Value	Description
height	Auto ,length ,%	Sets the height of an element
width	Auto, % ,length	Sets the width of an element
max-height	None ,length,%	Sets the maximum height of an element
max-width	None, length, %	Sets whether or not to show empty cells in a table
min-height	Length ,%	Sets the minimum height of an element
min-width	Length ,%	Sets the minimum width of an element

Example: - Set the height and width of image using pixel value.

```
<html>
<head>
<style type="text/css">
        img.normal
        img.normal {
        height: auto; width: auto
         }
         img.big {
         height: 160px; width: 160px
         }
        img.small {
        height: 30px; width: 160px
         }
                        <img class="normal" src="logocss.gif" width="95" height="84" />
                        <br />
                        <img class="big" src="logocss.gif" width="95" height="84" />
                        <br />
                        <img class="small" src="logocss.gif" width="95" height="84" />
                </body>
                </html>
```

Output: -

CSS Pseudo-classes

CSS pseudo-classes are used to add special effects to some selectors.

Pseudo Class	Description
Active	Adds special style to an activated element
Hover	Adds special style to an element when you mouse over it
Link	Adds special style to an unvisited link
Visited	Adds special style to a visited link

A link that is active, visited, unvisited, or when you mouse over a link can all be displayed indifferent ways in a CSS-supporting browser:

```
a:link {color: #FF0000}      /* unvisited link */
a:visited {color: #00FF00} /* visited link */
a:hover {color: #FF00FF} /* mouse over link */
a:active {color: #0000FF} /* selected link */
```

Pseudo-classes and CSS Classes

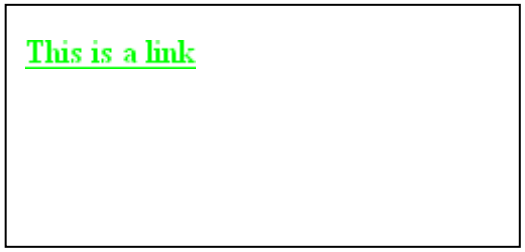
Pseudo-classes can be combined with CSS classes:

Example: - Example to demonstrate different hyperlink color

```
<html>
<head>
<style type="text/css">
    a:link {color:
    #FF0000}
    a:visited {color:
    #00FF00}
```

```
        a:hover {color:
        #FF00FF}a:active
        {color: #0000FF}
    </style>
    </head>
    <body>
    <p><b><a href="default.asp" target="_blank">This is a link</a></b></p></body>
    </html>
```

Output: -



What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

The syntax of pseudo-elements:

```
selector::pseudo-element {
  property: value;
}
```

The ::first-line Pseudo-element

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

<style>

p::first-line {

color: #ff0000;

font-variant: small-caps;

}

</style>

The ::first-letter Pseudo-element

The `::first-letter` pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all <p> elements:

<html>

<head>

```
<style>


p::first-letter {

color: #ff0000;

font-size: xx-large;

}

</style>

</head>

<body>

<p>You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!</p>

</body>

</html>
```

Pseudo-elements and HTML Classes

Pseudo-elements can be combined with HTML classes:

```
<style>

p.intro::first-letter {

color: #ff0000;

font-size: 200%;

}

</style>
```

Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

```
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}

p::first-line {
  color: #0000ff;
  font-variant: small-caps;
}
```

CSS - The ::before Pseudo-element

The `::before` pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before the content of each <h1> element:

```
h1::before {
  content: url(smiley.gif);
}
```

CSS - The ::after Pseudo-element

The `::after` pseudo-element can be used to insert some content after the content of an element.

The following example inserts an image after the content of each <h1> element:

```
h1::after {
  content: url(smiley.gif);
}
```

CSS - The ::marker Pseudo-element

The `::marker` pseudo-element selects the markers of list items.

The following example styles the markers of list items:

```
::marker {
  color: red;
  font-size: 23px;
}
```