

OBJECTTM
TECHNOLOGIES

Style sheet selectors

What will be covered

- Different CSS selectors
 - ID selector
 - class selector
 - universal selector
 - group selector
 - selection on parent-child relationship
 - selection for input elements
- CSS box model
- Displaying background images
 - Overlapping content
 - Columnar output

Different CSS selectors

OBJECT
Technology

- CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.
- There are several different types of selectors in CSS.
 1. CSS Element Selector
 2. CSS Id Selector
 3. CSS Class Selector
 4. CSS Universal Selector
 5. CSS Group Selector

CSS Element Selector

OBJECT

- The element selector selects the HTML element by name.

```
<style>
p{
    text-align: center;
    color: blue;
}
</style>
```

CSS Id Selector

OBJECT
TECHNOLOGY

- The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

- It is written with the hash character (#), followed by the id of the element.

```
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Javatpoint.com</p>
```

CSS Class Selector

- The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

```
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>

<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
```

CSS Class Selector

OBJECT

- CSS Class Selector for specific element

- If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

```
<head>
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>

<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
```

CSS Universal Selector

OBJECT

The universal selector is used as a wildcard character. It selects all the elements on the pages

```
<style>  
* {  
    color: green;  
    font-size: 20px;  
}  
</style>
```

CSS Group Selector

OBJECT

- The grouping selector is used to select all the elements with the same style definitions.
- Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```
<style>  
h1, h2, p {  
    text-align: center;  
    color: blue;  
}
```

CSS [*attribute*=*value*] Selector

OBJECT

- The [*attribute*=*value*] selector is used to select elements with the specified attribute and value.

Example

Style <a> elements with a target="_blank":

```
a[target=_blank] {  
    background-color: yellow;  
}
```

CSS selectors on parent-child relationship

OBJECT

- CSS can even select the HTML elements based on their parent, child and sibling realtionship.
- Following are the examples of CSS selectors

Sr No	Syntax	Example	Description
1	element element	div p	Selects all <p> elements inside <div> elements
2	element>element	div > p	Selects all <p> elements where the parent is a <div> element
3	element+element	div + p	Selects all <p> elements that are placed immediately after <div> elements
4	element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

CSS Selection for input elements

OBJECT

- There are some CSS selections mainly used for input elements in HTML form. Here are some examples

Sr No	Syntax	Example	Description
1	:optional	input:optional	Selects input elements with no "required" attribute
2	:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
3	:read-only	input:read-only	Selects input elements with the "readonly" attribute specified
4	:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
5	:required	input:required	Selects input elements with the "required" attribute specified
6	:valid	input:valid	Selects all input elements with a valid value

Other CSS selectors

OBJECT

Sr No	Syntax	Example	Description
1	:hover	a:hover	Selects links on mouse over
2	:last-child	p:last-child	Selects every <p> element that is the last child of its parent
3	:last-child	p:last-child	Selects every <p> element that is the last child of its parent
4	:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
5	:only-child	p:only-child	Selects every <p> element that is the only child of its parent

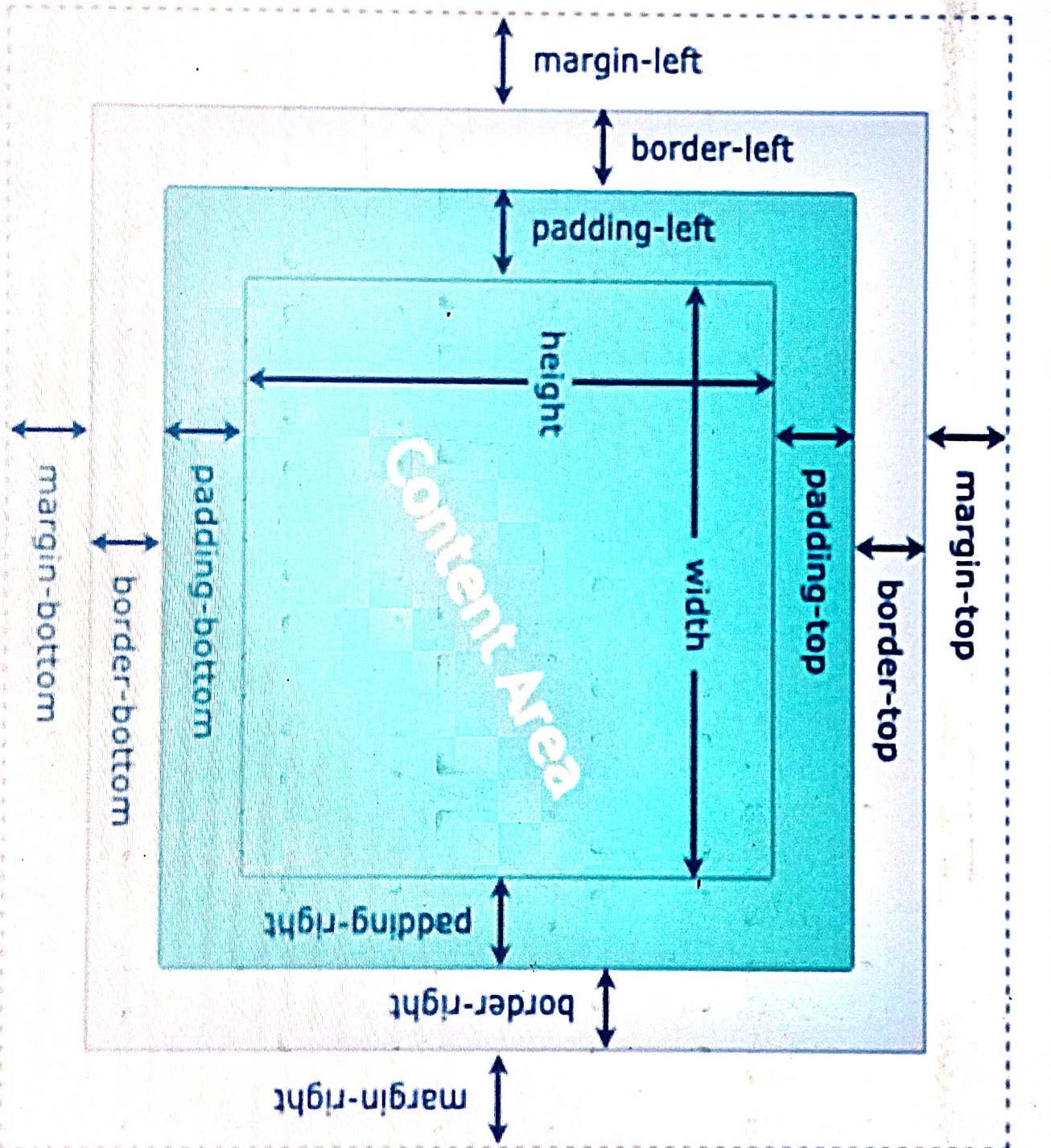
CSS box model

OBJECT
KNOWLEDGE

- When laying out a document, the browser's rendering engine represents each element as a rectangular box according to the standard CSS basic box model. CSS determines the size, position, and properties (color, background, border size, etc.) of these boxes.
- Every box is composed of four parts (or areas), defined by their respective edges: the content edge, padding edge, border edge, and margin edge.

CSS Box model

OBJECT



Displaying background images

OBJECT

- The background-image property is used to set an image as a background of an element. By default the image covers the entire element.

```
<style>
body {
background-image: url("paper1.gif");
margin-left:100px;
}
</style>
```

- By default, the background-image property repeats the background image horizontally and vertically.
- The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window.

Displaying background images

OBJECT

- The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

```
background: white url('good-morning.jpg');  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: center;
```

- The background shorthand property is a way to specify the values of multiple CSS background properties in a single declaration.
- The general syntax of background property is :

```
background: [background-image] [background-position] / [background-size]  
[background-repeat] [background-attachment] [background-origin]  
[background-clip] [background-color]
```

e.g.

```
body {  
background: url(photo.jpg) left top no-repeat #000;
```

Overlapping content

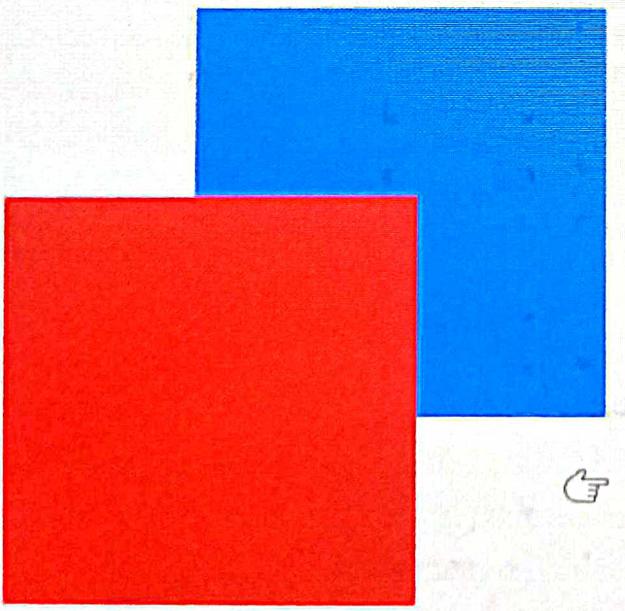
OBJECT

- Using CSS Z-Index property developer can stack elements onto one another. Z-Index can have a positive or a negative value.
- Elements are positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first.
- The position property specifies values like static, relative, fixed, absolute and sticky
- HTML elements are positioned static by default.
- Relatively positioned elements are placed relative to itself.
- For absolute position, the element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor.
- Elements with fixed position is positioned relative to the initial containing block established by the viewport

Overlapping content

OBJECT

```
html {  
    background-color: #f0f0f0;  
}  
  
body {  
    margin: 0;  
    padding: 0;  
    font-family: sans-serif;  
}  
  
div {  
    width: 200px;  
    height: 200px;  
    position: absolute;  
}  
  
.blue {  
    background: #007bff;  
    z-index: 1; /*The z-index of the blue box is lower than  
    that of the red box*/  
}  
  
.red {  
    background: #dc3545;  
    z-index: 2; /*The z-index of the red box is greater than tha  
    t of the blue box so it is shown after the blue box*/  
    left: 100px;  
    top: 100px;  
}  
  
</style>  
</head>  
<body>  
<div class="blue"></div>  
<div class="red"></div>  
</body>
```



Columnar output

OBJECT

- The float CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it.
- Absolutely positioned elements ignore the float property.
- When an element is floated, it is taken out of the normal flow of the document (though still remaining part of it). It is shifted to the left, or right, until it touches the edge of its containing box, or another floated element.
- Elements next to a floating element will flow around it. To avoid this, use the clear property.

Columnar output

OBJECT
METHOD

```
head
  style
    .left {
      float: left;
      background: cyan;
    }
    .right {
      float: right;
      background: pink;
    }
  
```

```
</style>
</head>
<body>

<section style="box-sizing: border-box; float:left; width:50%; ">
  <div class="left" style="width:50px; height:150px; border: 2px solid green;">1</div>
  <div class="left" style="width:50px; height:150px; border: 2px solid green;">2</div>
  <div class="right" style="width:50px; height:150px; border: 2px solid green;">3</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Morbi tristique sapien ac erat tincidunt, sit amet dignissim
  lectus vulputate. Donec id iaculis velit. Aliquam vel
  malesuada erat. Praesent non magna ac massa aliquet tincidunt
  vel in massa. Phasellus feugiat est vel leo finibus congue.</p>
</section>
</body>
```

Columnar output

OBJECT

- Output looks like this :

Aliquam vel malesuada erat. Praesent non magna ac massa aliquet tincident vel in massa. Phasellus feugiat est vel leo finibus congue.

10

Styles Sheet Selectors

In this chapter we will dive deep into CSS and learn more about different types of selectors that are available to define style rules. We will learn about CSS box model and frequently required designs for web pages in the form of columnar output etc.

Different CSS Selectors

CSS selectors help in selecting the different elements on the HTML page on which style rule can be applied. There are different types of selectors that are available which are based on name of tag, value of class attribute, value of id attribute, parent-child relationship, pseudo selectors etc.

Element Type Selectors

The most basic CSS selectors are Element Type Selectors. That's a fancy name for simply using an HTML tag, without the angle braces.

For example, if we wanted to make all paragraphs have green text, we would use the following CSS rule:

```
p { color: green; }
```

Universal Selector

Matches every element on the page.

For example, if we wanted every element to have a solid 1px wide border, we would use the following CSS rule:

```
* { border: 1px solid blue; }
```

Class Selectors

Match an element that has the specified class.

To match a specific class attribute, we always start the selector with a period, to signify that we are looking for a class value.

The period is followed by the class attribute value we want to match.

For example, if we wanted all elements with a class of "highlight" to have a different background color, we would use the following CSS rule:

```
.highlight { background-color: #ffffcc; }
```

If we want that all paragraph elements having class as "highlight" should be displayed with different color, following CSS rule can be useful.

Class Selectors

Match an element that has the specified class.

To match a specific class attribute, we always start the selector with a period, to signify that we are looking for a class value. The period is followed by the class attribute value we want to match.

For example, if we wanted all elements with a class of "highlight" to have a different background color, we would use the following CSS rule:

```
.highlight { background-color: #ffcccc; }
```

If we want that all paragraph elements having class as "highlight" should be displayed with different color, following CSS rule can be useful.

```
p.highlight { background-color: #ffcccc; }
```

Id Selectors

Match an element that has the specified id.

To match a specific id attribute, we always start the selector with a hash symbol (#), to signify that we are looking for an id value. The hash is followed by the id attribute value we want to match. Remember, we can only use the same id attribute value once, so the id selector will always only match one element in our document.

For example, if we wanted the element with an id of "content", we would use the following CSS rule:

```
#content { border: 2px solid green; }
```

Descendant Selectors

Match an element that is a descendant of another element.

This uses two separate selectors, separated by a space.

For example, if we wanted all emphasized text in our paragraphs to be green text, we would use the following CSS rule:

```
p em { color: green; }
```

Child Selectors

Match an element that is an immediate child of another element.

Child selectors

Match an element that is an immediate child of another element.

For example, if we wanted all emphasized text in our paragraph's to have green text, but not emphasized text in other elements, we would use the following CSS rule:

```
p>em { color: green; }
```

Adjacent sibling selectors

Match an element that is immediately after another element, but not a child of it.

For example, if we wanted all paragraphs that immediately followed an h4 to have green text, but not other paragraphs, we would use the following CSS rule:

```
h4 + p { color: green; }
```

Pseudo Selectors

An Aside About Link States

Anchor elements are special. You can style the <a> element with an Element Type Selector, but it might not do exactly what you expect. This is because links have different states, that relate to how they are interacted with. The four primary states of a link are: link, visited, hover, active.

Pseudo selectors come in different sizes and shapes. By far the most common pseudo selectors are used to style ~~old~~ links.

There are four different pseudo selectors to be used in conjunction with links:

```
:link
```

A link that has not been previously visited (visited is defined by the browser history)

```
:visited
```

A link that has been visited

```
:hover
```

A link that the mouse cursor is "hovering" over

```
:active
```

A link that is currently being clicked

```
a:link { color: red } /* unvisited links */  
a:visited { color: blue } /* visited links */
```

A link that has not been previously visited (visited is defined by the browser history)

:visited

A link that has been visited

:hover

A link that the mouse cursor is "hovering" over

:active

A link that is currently being clicked

```
a:link { color: red } /* unvisited links */
a:visited { color: blue } /* visited links */
a:hover { color: green } /* user hovers */
a:active { color: lime } /* active links */
```

CSS Properties

Here are some basic CSS properties to work with.

Font properties

Property	Description	Values
font	Sets all the font properties in one declaration	font-style, font-variant, font-weight, font-size/line-height, font-family, caption, icon, menu, message-box, small-caption, status-bar, inherit
font-family	Specifies the font family for text	family-name, generic-family, inherit
font-size	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %, inherit
font-style	Specifies the font style for text	normal, italic, oblique, inherit
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
font-weight	Specifies the weight of a font	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit
		Careful, many of these are not supported!

Property	Description	Values
font	Sets all the font properties in one declaration	font-style, font-variant, font-weight, font-size/line-height, font-family, caption, icon, menu, message-box, small-caption, status-bar, inherit
font-family	Specifies the font family for text	family-name, generic-family, inherit
font-size	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %, inherit
font-style	Specifies the font style for text	normal, italic, oblique, inherit
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
font-weight	Specifies the weight of a font	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit Careful, many of these are not supported!

Text properties

Property	Description	Values
color	Sets the color of a text	RGB, hex, keyword
line-height	Sets the distance between lines	normal, number, length, %
letter-spacing	Increase or decrease the space between characters	normal, length
text-align	Aligns the text in an element	left, right, center, justify
text-decoration	Adds decoration to text	none, underline, overline, line-through
text-indent	Indents the first line of text in an element	length, %
text-transform	Controls the letters in an element	none, capitalize, uppercase, lowercase

Text properties

Property	Description	Values
color	Sets the color of a text	RGB, hex, keyword
line-height	Sets the distance between lines	normal, number, length, %
letter-spacing	Increase or decrease the space between characters	normal, length
text-align	Aligns the text in an element	left, right, center, justify
text-decoration	Adds decoration to text	none, underline, overline, line-through
text-indent	Indents the first line of text in an element	length, %
text-transform	Controls the letters in an element	none, capitalize, uppercase, lowercase.

List properties

Property	Description	Values
list-style	Sets all the properties for a list in one declaration	list-style-type, list-style-position, list-style-image, inherit
list-style-image	Specifies an image as the list-item marker	URL, none, inherit
list-style-position	Specifies where to place the list-item marker	inside, outside, inherit
list-style-type	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, ugaritic, lower-roman, upper-roman, inherit

List properties

Property	Description	Values
<code>list-style</code>	Sets all the properties for a list in one declaration	<code>list-style-type, list-style-position, list-style-image, inherit</code>
<code>list-style-image</code>	Specifies an image as the list-item marker	URL, none, inherit
<code>list-style-position</code>	Specifies where to place the list-item marker	inside, outside, inherit
<code>list-style-type</code>	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit

Border properties

Property	Description	Values
<code>border</code>	Sets all the border properties in one declaration	<code>border-width, border-style, border-color</code>
<code>border-bottom</code>	Sets all the bottom border properties in one declaration	<code>border-bottom-width, border-bottom-style, border-bottom-color</code>
<code>border-bottom-color</code>	Sets the color of the bottom border	<code>border-color</code>
<code>border-style</code>	Sets the style of the bottom border	<code>border-style</code>

Border properties

Property	Description	Values
<code>border</code>	Sets all the border properties in one declaration	<code>border-width</code> , <code>border-style</code> , <code>border-color</code>
<code>border-bottom</code>	Sets all the bottom border properties in one declaration	<code>border-bottom-width</code> , <code>border-bottom-style</code> , <code>border-bottom-color</code>
<code>border-bottom-color</code>	Sets the color of the bottom border	<code>border-color</code>
<code>border-bottom-style</code>	Sets the style of the bottom border	<code>border-style</code>
<code>border-bottom-width</code>	Sets the width of the bottom border	<code>border-width</code>
<code>border-color</code>	Sets the color of the four borders	<code>color_name</code> , <code>hex_number</code> , <code>rgb_number</code> , <code>transparent</code> , <code>inherit</code>
<code>border-left</code>	Sets all the left border properties in one declaration	<code>border-left-width</code> , <code>border-left-style</code> , <code>border-left-color</code>
<code>border-left-color</code>	Sets the color of the left border	<code>border-color</code>
<code>border-left-style</code>	Sets the style of the left border	<code>border-style</code>
<code>border-left-width</code>	Sets the width of the left border	<code>border-width</code>
<code>border-right</code>	Sets all the right border properties in one declaration	<code>border-right-width</code> , <code>border-right-style</code> , <code>border-right-color</code>
<code>border-right-color</code>	Sets the color of the right border	<code>border-color</code>

border-right-color	Sets the color of the right border	border-color
border-right-style	Sets the style of the right border	border-style
border-right-width	Sets the width of the right border	border-width
border-style	Sets the style of the four borders	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit
border-top	Sets all the top border properties in one declaration	border-top-width , border-top-style , border-top-color
border-top-color	Sets the color of the top border	border-color
border-top-style	Sets the style of the top border	border-style
border-top-width	Sets the width of the top border	border-width
border-width	Sets the width of the four borders	thin, medium, thick, length, inherit

Box model, columnar output

Box model

Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS box model typically describes how these rectangular boxes are laid out on a web page. These boxes can have different properties and can interact with each other in different ways, but every box has a content area and optional surrounding padding, border, and margin areas.

The following diagram demonstrates how the width, height, padding, border, and margin CSS properties determines how much space an element can take on a web page.

↑ margin-top
↑ border-top

Box model, columnar output

Box model

Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS box model typically describes how these rectangular boxes are laid out on a web page. These boxes can have different properties and can interact with each other in different ways, but every box has a content area and optional surrounding padding, border, and margin areas.

The following diagram demonstrates how the width, height, padding, border, and margin CSS properties determines how much space an element can take on a web page.

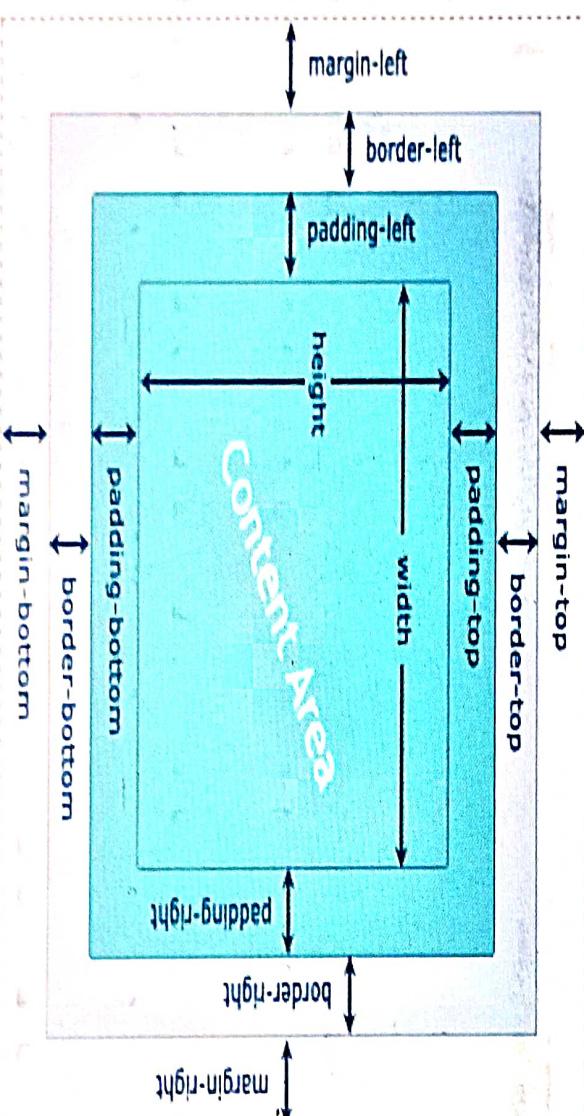


Fig: CSS BOX model

Padding is the transparent space between the element's content and its border (or edge of the box, if it has no border), whereas margin is the transparent space around the border.

Also, if an element has the background color it will be visible through its padding area. The margin area is always remain transparent, it is not affected by the element's background color, however, it causes the background color of the parent element to be seen through it.

Columnar output

The float CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it. Absolutely positioned elements ignore the float property. When an element is floated, it is taken out of the normal flow of the document (though still remaining part of it). It is shifted to the left or right, until it touches the edge of its container or another element.

Padding is the transparent space between the element's content and its border (or edge of the box, if it has no border), whereas margin is the transparent space around the border.

Also, if an element has the background color it will be visible through its padding area. The margin area is always remain transparent, it is not affected by the element's background color, however, it causes the background color of the parent element to be seen through it.

Columnar output

The float CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it. Absolutely positioned elements ignore the float property. When an element is floated, it is taken out of the normal flow of the document (though still remaining part of it). It is shifted to the left, or right, until it touches the edge of its containing box, or another floated element. Elements next to a floating element will flow around it. To avoid this, use the clear property.

```
<head>
<style>
.left {
  float: left;
  background: ■pink;
}
.right {
  float: right;
  background: ■cyan;
}
</style>
</head>
<body>
<section style="box-sizing: border-box; float:left; width:50%; ">
<div class="left" style="width:50px; height:150px; border: 2px solid ■green;">1</div>
<div class="left" style="width:50px; height:150px; border: 2px solid ■green;">2</div>
<div class="right" style="width:50px; height:150px; border: 2px solid ■green;">3</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Morbi tristique sapien ac erat tincidunt, sit amet dignissim
lectus vulputate. Donec id facilis velit. Aliquam vel
malesuada erat. Praesent non magna ac massa aliquet tincidunt
vel in massa. Phasellus feugiat est vel leo finibus congue.</p>
</section>
```

Output looks like this :

1
2

3
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi tristique sapien ac erat tincidunt, sit amet dignissim lectus vulputate. Donec id iaculis velit. Aliquam vel malesuada erat. Praesent non magna ac massa aliquet tincidunt vel in massa. Phasellus feugiat est vel leo finibus congue.

Assignments

1. Drop a shadow of the text Hello World (Hint use top, left, position, color css attributes)
2. Create a menu like following using CSS and list of hyperlinks:



3. Create a paragraph like following which displays paragraph like a transparent box.

This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box.

1
2
3

4. Create a page which display the paragraph given in the following HTML page:

4. Create a page which display the paragraph given in the following HTML page:

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

5. Give different style to the hyperlink as shown bellow:

[This link changes background-color](#)



[This link changes font-size](#)



[This link changes text-decoration](#)



[This link changes font-family](#)



[This link is fixed to the page](#)

