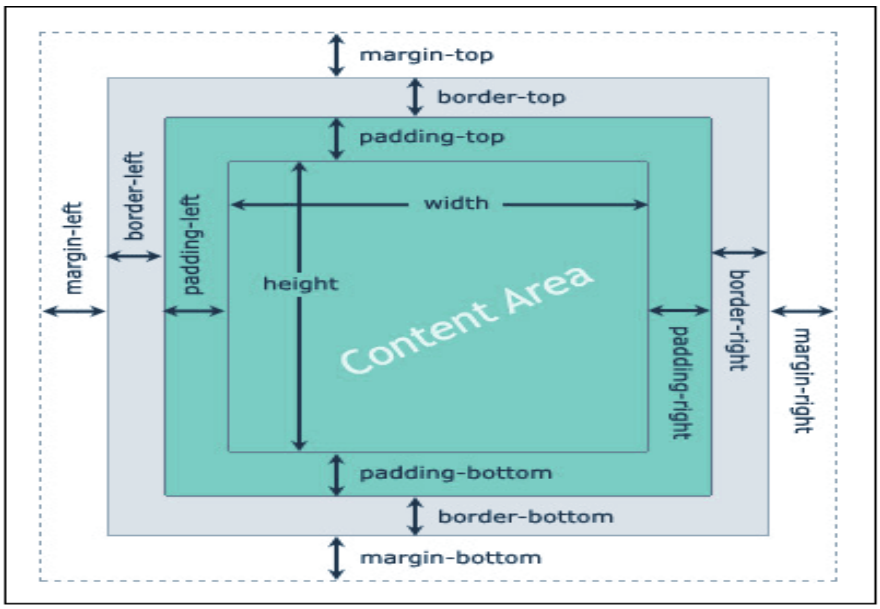
Web Designing
Overview of CSS

Box Model

- CSS treats an HTML document as a hierarchical tree of elements, where each element can have zero or more child elements arranged in an ordered way.
- The topmost element of this tree is called as the root element or the parent element.
- These elements display their content at a specific position, which is defined by using CSS properties.
- CSS converts the data of HTML elements in the form rectangular boxes, by using a layout model, called the box model, to set the design and layout of HTML documents.
- This means that the box model determines how HTML elements are displayed as boxes.
- The box model allows placing a border around the elements and also provides space between elements.

2

Box Model



Area of Box Model

Area	Description
Content area	Displays the content of a document, such as text and images. This is bounded by a rectangle, which is called as the content edge. Note that the content area always appears inside the padding area.
Padding area	Specifies the area around the content area. This is bounded by the padding edge. Outside the padding is the border area and the outside boundary of that area is the border edge. Finally, outside the border is the margin area whose outer edge is called the margin edge.
Border area	Specifies the area around the padding area. This is bounded by the border edge.
Margin area	Specifies the area around the border area. This is bounded by the margin edge.

Box Model

- A box model includes the following types of boxes :
  - **block-level** -- box Represents a box to show a paragraph
  - **line box** -- Represents a box to show a line of text
  - **inline-level box** -- Represents a box to show the words of a line
- **Note** - A block-level box can contain either other block-level boxes, such as a section containing paragraphs or a table containing rows, or line boxes, such as a paragraph containing lines of text.
- **Note** - A line box contains inline-level boxes, such as a line with words in different styles.
- **Note** - An inline-level box can contain either text with other inline-level boxes or a block-level box, such as a small table that is in inline format.

5

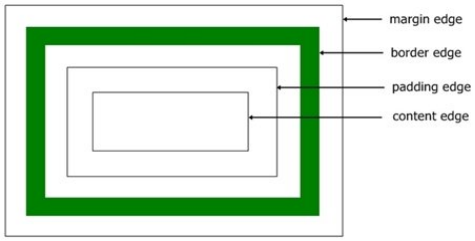
Aspects of Box Model

- Box dimensions
- The padding properties
- The border properties
- The margin properties
- The width and height properties
- Floating boxes
- Overflowing of box content
- The marquee properties
- Rotating boxes

6

Box Dimension

- All HTML elements in a box model are represented as rectangular boxes.
- The dimensions of the box model are calculated by using the height and width of the content area that gets applied to the element.
- Each box is associated with a content area and many optional areas, such as padding, border, and margin.
- The size of each area is specified by using the box model dimensions shown in the following figure:



7

Width and Height

- The width and height properties specify the width and height of the content area, padding area, or border area of a box.
- The syntax of these properties is given as follows:

  width/height: <length> | <percentage> | auto

- In the preceding syntax, the <length> specifies the width of the content area; <percentage> specifies the width of the content area in the percentage; and auto specifies that the width depends on the values of other properties. Cannot give negative value.
- Example:

  p { width: 100px; height: 50px; }

8

MinWidth and MinHeight

- The min-width and min-height properties are used to set the minimum width and height respectively, of a block-level element.
- The syntax of these properties is given as follows:

  min-width/min-height: <length> | <percentage> | inherit
- In the preceding syntax, <length> specifies a fixed minimum width height; <percentage> specifies a minimum value for width or height as a percentage of the corresponding dimension of the containing block; and none specifies no limit on the width or height of the box.

9

MaxWidth and MaxHeight

- The max-width and max-height properties are used to set the maximum width and height, respectively, of the block-level element.
- The syntax of these properties is given as follows:

  max-width/max-height: <length> | <percentage> | none
- In the preceding syntax, <length> specifies a fixed maximum width or height; <percentage> specifies a maximum value for width or height as a percentage of the corresponding dimension of the containing block; and none specifies no limit on the width or height of the box.
- Examples :

p { min-width: 10px; min-height: 10px; max-width: 100px; max-height: 100px; }

10

Box Dimension

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/css">
      p {min-width: 10px; min-height: 10px; max-width: 50em;
      max-height: 50em; border: 1em #888 inset;}
  </STYLE>
 </HEAD>
 <BODY>
   <P>demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
```

Box Dimension

```
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text </P>
 </BODY>  </HTML>
```

12

Padding

- Padding (space) in the box model specifies the distance between the border of an element and the content within it.
- The padding is affected by the background color of an element. The value of padding cannot be negative.
- The shorthand property in padding is used to change all the padding properties, such as padding-top, padding-bottom, padding-right, and padding-left at once.
- The syntax used to set padding properties is as follows:

  padding: value;

- Padding can be set according to the values describes in the table given here:
  - **Length** -- Specifies the fixed padding in the pt and em units.
  - **Percentage** -- Specifies padding in percentage with respect to the width of a parent block.
  - **Auto** -- Specifies the default padding from the top, bottom, left, or right direction.

13

Padding

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
      p{background-color:red;}
      p.padding{padding-top:50px;   padding-bottom:50px;
                padding-right:25px; padding-left:25px;}
    </STYLE>
  </HEAD>
  <BODY>
    <P class="padding">padding with top and bottom are 50px,
    right and left are 25px </P>
  </BODY>  </HTML>
```

14

Padding top

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
      p{border: double #FF0000;}
    </STYLE>
    <SCRIPT type="text/javascript">
      function changePadding(){
     document.getElementById("p1").style.paddingTop="5cm";
                                          }
    </SCRIPT>
  </HEAD>
```

15

Padding top

```
  <BODY>
    <P id="p1">This is a paragraph with top padding 5</P>
    <INPUT type="button" onclick="changePadding()"
     value="top padding" />
   </BODY>
</HTML>
```

16

Padding right

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
     <STYLE type="text/CSS">
        p {  border: double #FF0000; }
     </STYLE>
    <SCRIPT type="text/javascript">
       function rightPadding(){
  document.getElementById("p1").style.paddingRight="20cm";
                                    }
     </SCRIPT>
  </HEAD>
```

17

Padding right

```
  <BODY>
     <P id="p1">padding padding padding padding padding</P>
     <INPUT type="button" onclick="rightPadding()"
     value="Right padding" />
  </BODY>
</HTML>
```

18

Padding left

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
       p.padding {padding-left: 4cm;}
       p.padding2 {padding-left: 25%;}
    </STYLE>
  </HEAD>
  <BODY>
     <P class="padding">This is a paragraph with padding
     2cm</P>
     <P class="padding2">This is a paragraph with padding 25
    %</P>
</BODY>  </HTML>
```

19

Padding bottom

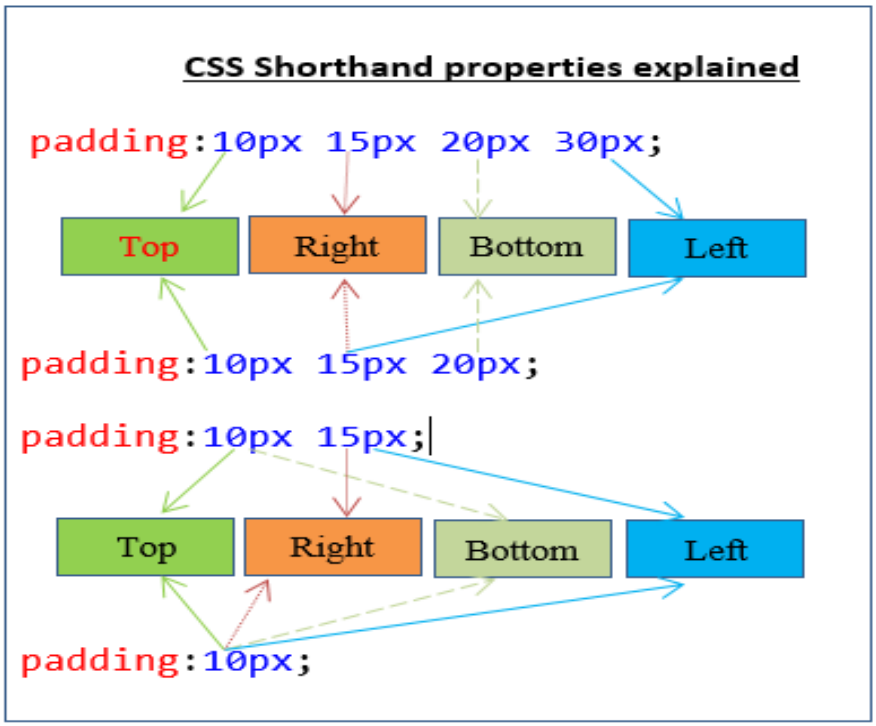
```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
     <STYLE type="text/CSS">
        p.padding {padding-bottom:2cm;}
        p.padding2 {padding-bottom:25%;}
    </STYLE>
 </HEAD>
 <BODY>
    <P class="padding">This is a paragraph with padding
     2cm</P>
    <P class="padding2">This is a paragraph with padding
    25%</P>
</BODY>  </HTML>
```

20

Padding shorthand

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
        p.p1 {padding:1cm;}
        p.p2 {padding:1cm 2cm;}
    </STYLE>
  </HEAD>
  <BODY>
      <P class="p1">This is a paragraph with equal padding on
      each side.</P>
      <P class="p2">This is a paragraph with top and bottom
      padding of 1cm and a left and right padding of 2cm.</P>
  </BODY>  </HTML>
```

21



Border

- The border properties specify the width, color, and style of the border area of a box. Syntax : **border 1px double value;**
- **Border width:** border-top-width, border-right-width, border-bottom-width, border-left-width
- **border-width** : The border width properties specify the width of the border area. Syntax : **border-width:value;**
- The value can be one of the following values:
  - Thin -- A thin border.
  - Medium -- A medium border.
  - Thick -- A thick border.
  - Length – Defines the thickness of the border.

23

Border

- If there is only one component value, it applies to all sides.
- If there are two values, the top and bottom borders are set to the first value and the right and left are set to the second.
- If there are three values, the top is set to the first value, the left and right are set to the second, and the bottom is set to the third.
- If there are four values, they apply to the top, right, bottom, and left, respectively.
- In the examples below, the comments indicate the resulting widths of the top, right, bottom, and left borders:
- h1 { border-width: thin } /* thin thin thin thin */
- h1 { border-width: thin thick } /* thin thick thin thick */
- h1 { border-width: thin thick medium } /* thin thick medium thick */

24

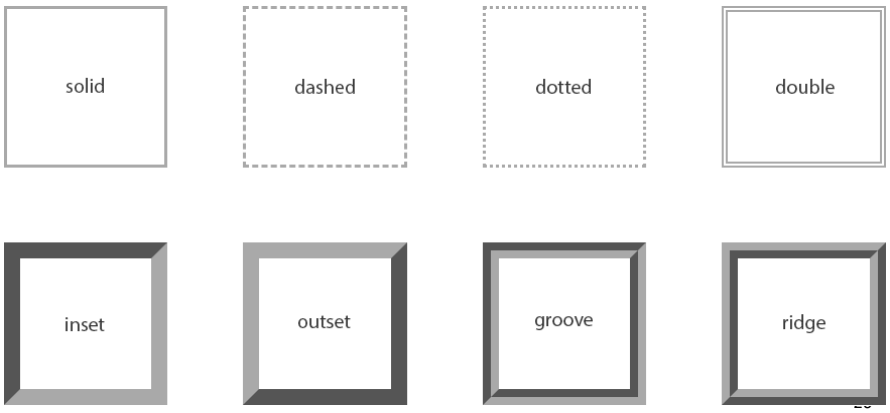
Border

- **Border color**: border-top-color, border-right-color, border-bottom-color, border-left-color
- **border-color** : The border color properties specify the color of a box's border.
- Syntax : border-color: value;
- Value: color | transparent | inherit
- If there is only one component value, it applies to all sides.
- If there are two values, the top and bottom borders are set to the first value and the right and left are set to the second.
- If there are three values, the top is set to the first value, the left and right are set to the second, and the bottom is set to the third.
- If there are four values, they apply to the top, right, bottom, and left, respectively.

25

Border

- **Border style:** border-top-style, border-right-style, border-bottom-style, border-left-style
- border-style : The border style properties specify the line style of a box's border (solid, double, dashed, etc.).



Types of Border

- **None** : No border; the computed border width is zero.
- **Hidden** : A line is drawn, but not visible. this can be handy for adding a little extra width to an element without displaying a border.
- **Dotted :** The border is a series of dots.
- **Dashed** : The border is a series of short line segments.
- **Solid :** The border is a single line segment.
- **Double** : The border is two solid lines. The sum of the two lines and the space between them equals the value of border-width.
- **Groove :** The border looks as though it were carved into the canvas.
- **Ridge**: The opposite of 'groove': the border looks as though it were coming out of the canvas.

27

Types of Border

- **Inset :** The border makes the box look as though it were embedded in the canvas.
- **Outset:** The opposite of 'inset': the border makes the box look as though it were coming out of the canvas.

28

Border example

```
<!DOCTYPE HTML>
<html>
 <head>
  <style type="text/css">
   .none {  border-style: none; }
  .solid {  border-style: solid; }
  .dashed {  border-style: dashed; }
  .dotted {  border-style: dotted;}
  .double {  border-style: double;}
  .mix {  border-style: double dashed dotted solid;}
  .groove {  border-style: groove;}
  .ridge {  border-style: ridge;}
```

29

Border example

```
  .inset {  border-style: inset;}
  .outset {  border-style: outset;}
  .hidden {  border-style: hidden;}
  .threevalues {  border-style:  groove hidden ridge;}

  /* Other styling */
  p {  font-family: Avenir;}
  </style>
 </head>
  <body>
```

30

Border example

```
<p class="none">No border</p>
<p class="solid">Solid border</p>
<p class="dashed">Dashed border</p>
<p class="dotted">Dotted border</p>
<p class="double">Double border</p>
<p class="mix">Mixed border</p>
<p class="groove">Groove border</p>
<p class="ridge">Ridge border</p>
<p class="inset">Inset border</p>
<p class="outset">Outset border</p>
<p class="hidden">Hidden border</p>
<p class="threevalues">Border defined by three values</p>
</body>  </html>
```

31

Border color

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
      p.border1{ border:groove #437845; }
      p.border2{ border-style:ridge; border-color: red black ; }
      p.border3{ border-style:double;border-width:4;
                 border-color:#123456 blue #894532;}
      p.border4{border-style:double; border-width:4;
       border-color:yellow green black rgb(250,0,255);}
     </STYLE>
  </HEAD>
```

32

Border color

```
<BODY>
    <P class="border1">This is a border with one color
     value</P>
    <P class="border2">This is a border with two color
    values</P>
    <P class="border3">This is a border with three color
    values</P>
    <P class="border4">This is a border with four color
    values</P>
 </BODY>
</HTML>
```

33

Border-radius

- The CSS border-radius property is used when we need to round the corner of the border.
- To use the border radius property, you need to specify the position of the corner of a box that is required to be in a round shape.
- The syntax of border-radius in CSS, is: **border-radius: x;**
- The value of x should be any of the following:
  - length - used to define how much the corner of the border has to be rounded. The default value is 0
  - % - used to define the same as done using length, but with the help of percentage. Therefore, 100% value is used to round the corner of the border for maximum
  - initial - used to set the default value
  - inherit - used to inherit the value from its parent

34

Border-radius

- We can also define the border-radius using multiple values.
  - If four values given, then first value applies to top left corner, second value applies to top right corner, third value applies to bottom right corner, and fourth value applies to bottom left corner
  - If three values given, then first value applies to top left, second value applies to top right and bottom left, whereas the third value applies to bottom right corner
  - If two values given, then first value applies to top left and bottom right, whereas the second value applies to top right and bottom left corner

35

Border Radius

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {border: 3px solid chocolate; padding: 20px;}
    p#one {border-radius: 10px 20px 30px 40px;}
    p#two {border-radius: 10px 20px 40px;}
    p#three {border-radius: 10px 40px;}
    p#four {border-radius: 10px;}
  </style>
</head>
```

36

Border Radius

```
<body>

  <p id="one">This is a para.</p>
  <p id="two">This is para two.</p>
  <p id="three">This is para four.</p>
  <p id="four">This is last para.</p>

</body>
</html>
```

37

Border image

- The CSS border-image property is used when we need to apply image as a border to an element.
- The syntax of border-image property in CSS, is:

border-image: source slice width outset repeat|initial|inherit;

- In above syntax:
  - source - used to define the path of the image
  - slice - used to define the way to slice the image
  - width - used to define the width of the border image
  - outset - used to define the amount to extend the area of the border image, beyond the border box
  - repeat - used to define how the image should be scaled and tiled like rounded, stretched, or repeated etc.

38

Border image

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE> Border Image </TITLE>
    <style>
      P{border:solid 10px;  border-image: url(1.png) 4 round;}
    </style>
  </HEAD>
  <BODY>
    <P> demo text demo text demo text demo text demo text demo
     text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
```

39

Border image

```
    text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text
    </P>
  </BODY>
</HTML>
```

40

Margin

- The blank area around the border of an element is called margin.
- Margin is used to create an extra space around an element. It is completely transparent and does not contain any background color.
- Margin is also used to determine spacing around different elements. The margin property is used to set all the sides of an element, such as top, right, bottom, and left.
- The syntax to set the margin property is as follows:

  p { margin: value; }

41

Possible values of Margin

Value	Description
Auto	Defines margin.
Length	Defines a fixed margin in px, pt, and cm. The default value of this property is 0px.
%	Defines margin in percentage. The value of margin is the height of the nearest block.
Inherit	Defines that margin should be inherited from the parent element.

42

Margin

- A box has four sides; therefore, the margin in a box is divided into the following four properties:
  - margin-top
  - margin-bottom
  - margin-right
  - margin-left
- The following code fragment shows the specifications of margin property in the box model:

#container{ margin-top: 7px; margin-left: 25%; margin-right: auto; margin-bottom: 45px; border: 2px solid black; }

- The top, right, bottom, and left values of a margin can be changed independently by using different properties, such as margin-top, margin-right, margin-bottom, and margin-left.

43

Margin

- A margin shorthand property is used to change all the properties of margin at once. The values of margin property ranges between one and four. Let's take a look at each case one by one.

1. If CSS margin property contains four values like this:

   margin: 25px 50px 75px 100px;

- Then, these four values represents: top margin is 25px, right margin is 50px, bottom margin is 75px, left margin is 100px

2. If CSS margin property contains three values like this:

   margin: 25px 50px 75px;

- Then, these three values represents: top margin is 25px,right and left margins are 50px, bottom margin is 75px

44

Margin

3. If CSS margin property contains two values like this:

   margin: 50px 100px;

- Then, these two values represents: top and bottom margins are 50px, right and left margins are 100px

4. If CSS margin property contains only one value like this:

   margin: 100px;

- Then, the above value represents: all four margins are 100px

45

margin

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <STYLE type="text/CSS">
        p{background-color: red;}
        p.margin{margin:200px 250px;}
     </STYLE>
  </HEAD>
   <BODY>
      <P class="margin">This is a margin with top and bottom
       have 200 px and right and left have 250px</P>
   </BODY>
</HTML>
```

46

margin top

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>MarginTop</TITLE>
  </HEAD>
   <BODY style="background:green;">
      <P style="background:orange;padding: 5cm;border-width:
      1em;border-style: solid;border-color: aqua;margin-top:5cm;">
         This is a margin with 5cm from the top
      </P>
   </BODY>
</HTML>
```

47

margin right

```
<!DOCTYPE HTML>
<HTML>
   <HEAD>
      <TITLE>MarginRight</TITLE>
   </HEAD>
    <BODY style="background:green;">
       <P style="background:orange;padding: 5cm;border-width:
        1em;border-style:  solid;border-color: aqua;margin-right:
        10cm;">
          This is a margin with 10cm from the right
        </P>
   </BODY>
</HTML>
```

48

margin bottom

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>MarginBottom</TITLE>
  </HEAD>
  <BODY style="background:green;">
    <P style="background:orange;padding: 5cm;border-width:
    1em;border-style: solid;border-color: aqua;margin-bottom:
    10cm;">
      This is a margin with 10cm from the bottom
    </P>
  </BODY>
</HTML>
```

49

margin left

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>MarginLeft</TITLE>
  </HEAD>
  <BODY style="background:green;">
    <P style="background:orange;padding: 5cm;border-width:
    1em;border-style:solid;border-color: aqua;margin-left:
    5cm;">
      This is a margin with 5cm from the left
    </P>
  </BODY>
</HTML>
```

50

margin shorthand

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE> Marginshorthand </TITLE>
  </HEAD>
  <BODY style="background:green;">
    <P style="background:orange;padding: 5cm;border-width:
     1em;border-style: solid;border-color: aqua;margin:5cm 6cm
      5cm 6cm;">
        This is a paragraph with four specified margin values
    </P>
  </BODY>
</HTML>
```

51

Floating Box

- A floating box is a box that can display the content of an element in left or right direction with respect to the content of another element.
- For example, you can display a box containing the content of a paragraph in left direction and an image in the right direction. You can float an element by using the float property.
- The syntax to use the float property is given as follows :

  float: left | right | none | <page-floats>
- Example:

  img { float: left; }

52

Floated Box

```
<!DOCTYPE HTML>
<HTML>
 <HEAD>
  <TITLE>Float example</TITLE>
  <STYLE type="text/css">
    IMG { float: left; border: solid lime; }
    P{ margin: 2em;border: solid red; }
  </STYLE>
 </HEAD>
 <BODY>
  <P><IMG src= "photo_1.gif" alt="floated image">
   demo text demo text demo text demo text demo text demo text
   demo text demo text
```

53

Floated Box

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text

54

Floated Box

demo text demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text demo text demo text demo text demo text demo text demo text

demo text demo text demo text </P>

</BODY>

</HTML>

55

Line Box Model

- CSS3 introduces a new model, called the line box model to enhance the box model.
- The line box model helps to display elements and text in line with each other.
- It also enables you to display first line of an element in a specified format or to display the first character of an element in big font size.

56

Line Box Model

Property	Description
alignment-adjust	Allows you to specify alignment points for elements
alignment-baseline	Specifies the alignment of an inline-level element with respect to its parent
baseline-shift	Allows you to reposition the dominant-baseline with respect to other dominant-baseline
dominant-baseline	Specifies a scaled-baseline-table
drop-initial-after-adjust	Specifies an alignment point for the initial letter box with respect to the primary connection point
drop-initial-after-align	Specifies an alignment line within the initial line box that is used at the primary connection point with the initial letter box
drop-initial-before-adjust	Specifies an alignment point of the drop initial for the secondary connection point

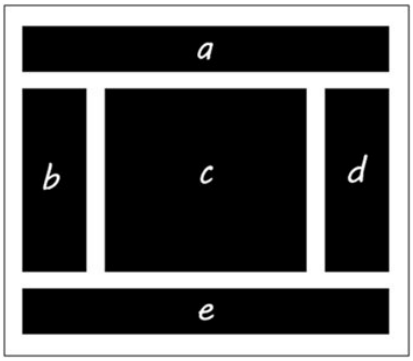
57

Line Box Model

Property	Description
drop-initial-before-align	Specifies an alignment line within the initial line box that is used at the secondary connection point with the initial letter box
drop-initial-size	Specifies a value to show the size of the initial letter
drop-initial-value	Specifies a value to activate a drop-initial effect
inline-box-align	Specifies the alignment of a line of a multi-line inline block with the previous and next inline elements within a line
line-stacking	Specifies a shorthand property for setting the line-stacking-strategy, line-stacking-ruby, and line-stacking-shift properties
line-stacking-ruby	Specifies the line stacking method for elements, which are ruby annotation elements
line-stacking-shift	Specifies the line stacking method for elements, which are baseline-shifted elements
line-stacking-	Specifies the height of the text content area in an inline box

Template Layout Model

- CSS3 introduces a new layout model, called the template layout model, which allows you to present the content contained in elements into slots.
- The slots are the placeholders that can be created by declaring a grid structure using certain alphabetical characters as shown in the following figure (Using Alphabets to Create Slots in a Template):



59

Template Layout Model

- Using the template layout model, you can divide the layout of a Web page or a form in following two parts :
  - Grid of the page or window
  - Fonts, indents, and colors of the text and other objects
- The template layout model defines a layout policy, called template-based positioning, to give an invisible grid to an element for aligning with other elements.
- A template does not allow elements to overlap and provides layouts with different widths, heights, and alignments.
- An element mapped to its slot in a template by using the position property.
- The template itself is specified by using the display property for remapping of some elements.

60

Template Layout Model

- The display property defines, how an element should rendered on the web page. The display: grid declaration converts the element into a grid container.
- The grid-template-areas property defines the area of a/multiple grid item(s) in grid layout.
- The grid-area property defines the location and size of a grid item in grid layout.

61

Template Layout Model

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .xContainer {display: grid; gap: 4px;
      grid-template-areas: 'head head head head head' 'nav con con
      con oth' 'foot foot foot foot foot';}
    .xContainer > div {background-color: black; color: white;
                       padding: 10px; text-align: center;}
    #xHeader {grid-area: head;}
    #xMenu {grid-area: nav; min-height: 300px;}
    #xContent {grid-area: con; min-height: 300px;}
    #xOther {grid-area: oth; min-height: 300px;}
    #xFooter {grid-area: foot;}
  </style>
</head>
```

62

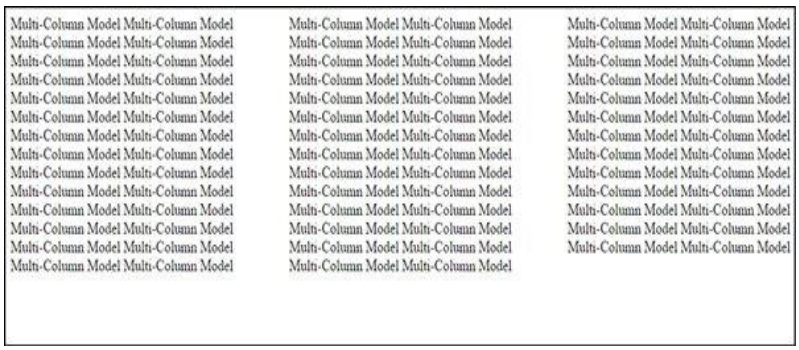
Template Layout Model

```
<body>
  <div class="xContainer">
    <div id="xHeader">Header</div>
    <div id="xMenu">Menu</div>
    <div id="xContent">Content</div>
    <div id="xOther">Other</div>
    <div id="xFooter">Footer</div>
  </div>
</body>
</html>
```

63

Multi-Column Model

- To display your Web content in multiple columns like the columns of a newspaper.
- CSS enables you to show your element content in multiple columns.
- Following figure shows the content of a Web page in multiple columns :



64

Multi-Column Model Properties

- **columns** - defines column-width and column-count at once
- **column-width** - defines same width to all columns of an element
- **column-count** - divides an element to specified number of columns
- **column-rule** - defines width, style, and color of column rule, at once
- **column-rule-width** - defines the width of rule created between columns
- **column-rule-style** - defines the style of rule created between columns
- **column-rule-color** - defines the color of rules created between columns
- **column-span** - defines the whether an element should span across all columns or not
- **column-fill** - fills each columns with equally distributed contents or not

65

Multi-Column Model

- CSS provides the following properties to show the content in multiple columns :
- **columns** - defines column-width and column-count at once
- **column-width** - defines same width to all columns of an element
- **column-count** - divides an element to specified number of columns
- **column-rule** - defines width, style, and color of column rule, at once
- **column-rule-width** - defines the width of rule created between columns
- **column-rule-style** - defines the style of rule created between columns
- **column-rule-color** - defines the color of rules created between columns

66

Column Width & Column Count

```
<!<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE> Column Width Count </TITLE>
  </HEAD>
  <BODY">
    <P style="background:#0067FF;border-width: 1em;border-style: outset;border-color: red;column-count: 3;column-width: 200px;">
    demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo
```

67

Column Width & Column Count

text demo text

demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo

text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo

demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo

text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo

demo text demo text demo text demo text demo text demo

68

Column Width & Column Count

```
        text demo text demo text demo text demo text demo text
        demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text
    </P>
  </BODY>
</HTML>
```

69

Column Gap & Column rule

```
<!DOCTYPE HTML>
<HTML>
   <HEAD>
      <TITLE> Column Gap Rule </TITLE>
   </HEAD>
   <BODY">
        <P style="background:#0067FF;border-width: 1em;border-
         style: outset;border-color: red;column-count: 2;
          column-width: 200px;column-gap:100px;column-rule:
          20px yellow double;">
        demo text demo text demo text demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
```

70

Column Gap rule

```
        text demo text demo text        demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text        demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text        demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text        demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text        demo text demo text demo
        text demo text demo text demo text demo text demo text
    </P>
</BODY>   </HTML>
```

71

Column Fill and Column Span

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>Column demo</TITLE>
    <STYLE type="text/css">
        h1 {column-span: all ; background: silver }
        body {background:purple; border: lime 10px solid;
                column-count: 3;
                column-rule: 10px solid cyan;
                column-fill: balanced;}
     </STYLE>
   </HEAD>
```

72

Column Fill and Column Span

<BODY>

demo text demo text

<H1>This heading for all columns</H1>

demo text demo

Column Fill and Column Span

text demo text demo

Column Fill and Column Span

```
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text demo text demo text demo   text
        demo text demo text demo text demo text demo text demo
        text demo text demo text demo text demo text demo text
        demo text demo text demo text demo text demo text demo
        text demo text demo text demo text demo text
    </BODY>
</HTML>
```

75

Column Break

```
<!DOCTYPE HTML>
<HTML>
   <HEAD>
     <TITLE> Column Break </TITLE>
   </HEAD>
   <STYLE type="text/css">
      #wrapper{ width: 30em; height: 20em;
                  border: 3px double magenta; margin: 80px auto;
                  column-count:5;
                  column-rule: solid   }
      .box{ width: 100px; height: 100px;  background: lime;
            margin: 1px; column-break-after: always;    }
   </STYLE>
```

76

Column Break

```
<BODY>
  <DIV id="wrapper">
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo demo text demo text demo text
    demo text demo text demo text demo text text</DIV>
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text demo text demo text demo text </DIV>
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text
    demo text demo text </DIV>
```

77

Column Break

```
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text </DIV>
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text </DIV>
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text </DIV>
  <DIV class="box">demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text demo text demo
    text demo text demo text demo text demo text demo text
    demo text demo text demo text demo text</DIV>
```

78

Column Break

```
    </DIV>
  </BODY>
</HTML>
```

Note : the content is divided into 5 columns. However a column break is inserted always at the start of each DIV

79

page break

```
<!DOCTYPE HTML>
<HTML>
   <HEAD>
      <TITLE> Page Break </TITLE>
    </HEAD>
    <STYLE type="text/css">
        body{border: 10px blue solid;}
        p { column-count: 4;column-rule: 10px red
        solid;break-after: page}
         H1 {break-before: page;break-after: page }
    </STYLE>
    <BODY>
       demo text demo text demo text demo text demo text demo
```

80

page break

text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text<P>paragraph paragraphparagraph paragraph </P><P>paragraph paragraph

81

page break

paragraph paragraphparagraph </P><H1>This is heading 1 content</H1>demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text demo text <P>paragraph paragraph</P>

82

83