

Front End Web Development

7/18/2022



CSS Padding

- Padding is used to create space around an element's content, inside of any defined borders.
- If the padding property has one value:
 - padding: 50px;
 - all four paddings are 50px

19

```
padding: 50px;
```



CSS Padding

- CSS has properties for specifying the padding for each side of an element:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left

```
87  div {  
88      padding-top: 50px;  
89      padding-right: 30px;  
90      padding-bottom: 50px;  
91      padding-left: 80px;  
92  }
```



CSS Padding-Shorthand Property

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

```
87  div {  
88      padding: 25px 50px 75px 100px;  
89  }
```



Padding

- 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

```
87  div {  
88      padding: 25px 50px 75px;  
89  }
```



Padding

- If the padding property has two values:
 - padding: 25px 50px;
 - top and bottom paddings are 25px
 - right and left paddings are 50px

```
87  div {  
88      padding: 25px 50px;  
89  }
```



Exercise

- Add padding to each of your divs
 - Use one padding value on one div
 - Use two padding values on one div
 - Use three padding values on one div
 - Use four padding values on one div



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).



CSS Margins

- If the margin property has one value:
 - margin: 25px;
 - all four margins are 25px

```
87  p {  
88      margin: 25px;  
89  }
```



Margin - Individual Sides

- CSS has properties for specifying the margin for each side of an element:
 - margin-top
 - margin-right
 - margin-bottom
 - margin-left

```
87  p {  
88      margin-top: 100px;  
89      margin-bottom: 100px;  
90      margin-right: 150px;  
91      margin-left: 80px;  
92  }
```



Margin - Shorthand Property

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

```
87 p {  
88     margin: 25px 50px 75px 100px;  
89 }
```



Margin

- 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

```
87  p {  
88      margin: 25px 50px 75px;  
89  }
```



Margin

- If the margin property has two values:
 - margin: 25px 50px;
 - top and bottom margins are 25px
 - right and left margins are 50px

```
87  p {  
88      margin: 25px 50px;  
89  }
```



Exercise

- Add margin to each of your divs
 - Use one margin value on one div
 - Use two margin values on one div
 - Use three margin values on one div
 - Use four margin values on one div



CSS Text Decoration

- The text-decoration property is used to set or remove decorations from text.
- These text-decoration values are used to decorate text

```
38 h1 {  
39     text-decoration: overline;  
40 }  
41  
42 h2 {  
43     text-decoration: line-through;  
44 }  
45  
46 h3 {  
47     text-decoration: underline;  
48 }
```



Exercise

- Add a different text decoration to each of your divs



CSS 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

```
39  p.uppercase {  
40      text-transform: uppercase;  
41  }  
42  
43  p.lowercase {  
44      text-transform: lowercase;  
45  }  
46  
47  p.capitalize {  
48      text-transform: capitalize;  
49  }
```



Exercise

- Add a different text transformation to each of your divs



CSS Text Spacing

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

```
40 p {  
41     text-indent: 50px;  
42 }
```



CSS Text Spacing

- 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

```
40  h1 {  
41      letter-spacing: 3px;  
42  }  
43  
44  h2 {  
45      letter-spacing: -3px;  
46  }
```



Exercise

- Add text indentation and letter spacing to each one of your divs



CSS Flexbox Layout

Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure.



Flexbox Layout

- The Flexbox Layout aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is dynamic.
- Flex layout gives the container the ability to alter its items width, height, and order to best fill the available space
- Flex layout is direction-agnostic (its free from any directional constraints) whereas block layout vertically biased and inline layout is horizontally biased



Flexbox

To start using the Flexbox model, you need to first define a flex container.

A flex container with three flex items:

```
14 <div class="flex-container">
15   <div>1</div>
16   <div>2</div>
17   <div>3</div>
18   <div>4</div>
19 </div>
```



Flexbox

The flex container becomes flexible by setting the display property to flex:

```
16  .flex-container {  
17      display: flex;  
18  }
```



Styling divs inside of divs

Use a “>” symbol between the class and div element to style divs inside of divs

```
8  .flex-container > div {  
9      background-color: ■ #f1f1f1;  
10     margin: 10px;  
11     padding: 20px;  
12     font-size: 30px;  
13 }
```



Exercise

1. Create 4 flex containers with 4 items inside of them
2. Style the flex containers and the divs inside of them



The flex-direction Property

The flex-direction property defines in which direction the container wants to stack the flex items.

The column value stacks the flex items vertically (from top to bottom):

```
1 .flex-container {  
2     display: flex;  
3     flex-direction: column;  
4     background-color: blue;  
5 }
```



The flex-direction Property

The column-reverse value stacks the flex items vertically (but from bottom to top):

```
1 .flex-container {  
2     display: flex;  
3     flex-direction: column-reverse;  
4     background-color: blue;  
5 }
```



The flex-direction Property

The row value stacks the flex items horizontally (from left to right):

```
1 .flex-container {  
2     display: flex;  
3     flex-direction: row;  
4     background-color: blue;  
5 }
```



The flex-direction Property

The row-reverse value stacks the flex items horizontally (but from right to left):

```
1  .flex-container {  
2      display: flex;  
3      flex-direction: row-reverse;  
4      background-color: blue;  
5  }
```



Exercise

1. Change the flex-direction of your flex containers
2. Use
 - a. Row
 - b. Row-reverse
 - c. Column
 - d. Column Reverse



The flex-wrap Property

The flex-wrap property specifies whether the flex items should wrap or not.

The wrap value specifies that the flex items will wrap if necessary:

```
1 .flex-container {  
2     display: flex;  
3     flex-wrap: wrap;  
4     background-color: blue;  
5 }
```



The flex-wrap Property

The nowrap value specifies that the flex items will not wrap:

```
1 .flex-container {  
2     display: flex;  
3     flex-wrap: nowrap;  
4     background-color: blue;  
5 }
```



The flex-wrap Property

The wrap-reverse value specifies that the flexible items will wrap if necessary, in reverse order:

```
1 .flex-container {  
2     display: flex;  
3     flex-wrap: wrap-reverse;  
4     background-color: blue;  
5 }
```



Exercise

1. Style the flex container and the divs
2. Use:
 - a. Wrap
 - b. Nowrap
 - c. Wrap-reverse

