`Day-3: CSS Box Model & CSS Combinators`

CSS Box-Model:

- The CSS box model is a container that contains multiple properties including borders, margin, padding, and the content itself. It is used to create the design and layout of web pages. According to the CSS box model, the web browser supplies each element as a square prism.

Visualizing Box-model

- In the above figure, we have
  - one parent box(red-box)
  - four child boxes (blue-box)

Pre-requisites of building box

- Height
- Width
- Border-color

Height & Width

- The CSS `height` and `width` properties are used to set the height and width of an element.
- The `height` and `width` properties may have the following values:
  - `auto` - This is default. The browser calculates the height and width
  - `length` - Defines the height/width in px, cm etc.
  - `%` - Defines the height/width in percent of the containing block

Block vs inline elements

Block	Inline
The element will start on a new line and occupy the full width available. And you can set width and height values.	The element doesn't start on a new line and only occupy just the width it requires. You can't set the width or height.
Block elements accept all the margins	Inline elements accept only left and right margins and do not accept top and

Various block and inline elements

Properties of the Box Model

Content

- The content area consists of content like image, text, or other forms of media content. The height and width properties help to modify the box dimensions.

Padding

- The padding area is the space around the content area and within the border-box. It can be applied to all sides of the box or to the specific, selected side(s) - top, right, bottom, and/or left.

Border

- The border area surrounds the padding and the content, and can be applied to all the sides of the box or to selected side(s) - top, right, bottom, and/or left.

Margin

- The margin area consists of space between the border and the margin. The margin does not possess its own background color and is completely transparent. It shows the background color of the element, like the body element.

What is Margin in CSS?

- Margin is a CSS property that defines the empty space around an HTML element. Margins ensure that the specified region around an element remains unoccupied by any neighboring element.

Syntax

There are four margin properties, one for each side of the HTML element box.

- **margin-top:** adds margin space on top of the element.
- **margin-right:** adds margin space on the right of the element.
- **margin-bottom:** adds margin space on the bottom of the element.
- **margin-left:** adds margin space on the left of the element.

- In above example, we are setting a margin of 20px on the top and left sides and also a margin of 40px on the right and bottom side.

CSS Margin Shorthand Property

- **Single Value (margin:20px)** :
  - The single value is set as margin on all four sides.

- **Two Values (margin:4px 5px):**
  - The first value is set as vertical margin (top & bottom) while the second is set as horizontal margin (right & left).

- **Three Values (margin:4px 5px 2px):**

- The first value is set as top margin, second as horizontal margin (right & left) while the third is set as bottom margin.

- **Four values (margin:4px 5px 2px 1px):**
  - The values are assigned to margins on sides starting from top and moving in a clockwise direction, i.e. the four values are set as top, right, bottom and left margin respectively.

Negative margin:

- Positive margin values move the content towards the inner side of its position or page. But if we take negative margin values, it moves the content towards outside of its position or page. The margin property in html gives space around the outermost element's content of the box-like structure

- A funny example of negative margin

CSS Combinators:

- The CSS combinators represent the relationship between two selectors.
- The CSS selectors are the patterns that can be used for styling the particular HTML element. Sometimes, it is possible that there is more than one simple selector, and to combine the multiple simple selectors, we use the combinators.

Why use CSS Combinator ❓

- Learning about combinators makes you **better at writing CSS** and **helps you to avoid excess CSS code.**

- Combinators can also help you **pinpoint the section** or **part of HTML** you want to style with high accuracy because they are based on the **relationship between the selectors.**

Types of Combinators in CSS

The combinators are of 4 types, which are given below:

1. `Descendant selector ( ) (space)`
2. `Child selector (>)`
3. `General sibling selector (~)`
4. `Adjacent sibling selector (+)`

Descendant Selector in CSS

- **Descendant Meaning:** A person who is related to you and who lives after you, such as your child or grandchild
- Min 2 , can be greater than 2
- `Syntax`

```
selector1 selector2 selector3... {
  // style properties
}
```

Practical example:

- Look at Amitabh's family tree
  - Number of Descendants for Harivansh Rai Bachan - 15
  - Number of Descendants for Amitabh Bachan - 7
  - Number of Descendants for Abhishek Bachan - 1 (Aaradhya)
  - Number of Descendants for Sweta Nanda - 2

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
    /* It will select all p's which are descendants of div */
    background-color: teal;
}
</style>
</head>
<body>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <span><p>Paragraph 3 in the div.</p></span>
</div>
<p>Paragraph 4. Not in a div, not a descendant</p>
<p>Paragraph 5. Not in a div, not a descendant</p>

</body>
</html>
```

- In the above figure div, p will select all p's which are descendants of div

Child Selector

- The child selector uses the greater than sign `(>)` to separate the elements. The child selector is used when we want to apply the styling properties to the immediate child/children of the particular HTML element.
- This combinator is quite strict than the descendant selector and the styling properties are acquired only when the second selector is the direct child of the first one.

- Look at Amitabh's family tree
  - Number of Childs for Harivansh Rai Bachan - 2
  - Number of Childs for Amitabh Bachan - 2
  - Number of Childs for Abhishek Bachan - 1 (Aaradhya)
  - Number of Childs for Sweta Nanda - 2

- Live code: Codepen

- In the above figure `div>p` will select all p's which are `childs` of div

General Sibling Selector:

- **Sibling meaning:** A brother or sister from the same parent
- The general sibling selector is used when the user wants to set the CSS properties for the elements that are the siblings of each other even if they are not the immediate ones.
- This selector is used when we have to set the styling properties of the elements that have the same parent element. This selector can be separated by adding the (~) sign between them.

- Look at Amitabh's family tree
  - Number of general siblings for Amitabh Bachan - 1 (Ajitabh Bachan)
  - Number of general siblings for Abhishek Bachan - 1 (Sweta Nanda)

- Number of general siblings for Bhim Bachan - 3
- Number of general siblings for Aaradhy - 0

Remember that siblings will be of the same generation

- Live code: Codepen

- In the above figure `h1 ~ p` will select all p's which are siblings of h1

Adjacent / Immediate sibling Selector in CSS

- The adjacent sibling selector is used when we want to apply the CSS property or styling to the adjacent sibling of any element.
- The siblings should have the same parent element and also the second element must be the immediate follower of the first element.
- The selectors are separated by adding the (+) sign between the separators.

- Look at Amitabh's family tree
  - Number of immediate siblings for Amitabh Bachan - 1 (Ajitabh Bachan)
  - Number of immediate siblings for Abhishek Bachan - 1 (Sweta Nanda)
  - Number of immediate siblings for Bhim Bachan - 1 (Namrata Bachan)

- Number of immediate siblings for Navya Naveli - 1

Remember that siblings will be of the same generation

Overview of all selectors:

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects all <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every <ul> element that are preceded by a <p> element

More Selectors:

CSS *element,element* Selector

Select and style all <h2> elements AND all <p> elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
h2, p {
  background-color: yellow;
}
</style>
</head>
<body>

<h1>Demo of the element, element selector</h1>

<h2>Welcome to My Homepage</h2>

<div>
  <p>My name is Donald.</p>
  <p>I live in Duckburg.</p>
</div>
```

```
<p>My best friend is Mickey.</p>

</body>
</html>
```

CSS Attribute Selectors

- The `[attribute]` selector is used to select elements with a specified attribute.

Live code : Link

Types of attribute selector

	Selector	Example	Description
Presence & Value	[attr]	a[title]	Matches elements with an *attr* attribute (whose name is the value in square brackets).
	[attr=value]	a[href="https://example.com"]	Matches elements with an *attr* attribute whose value is exactly *value* — the string inside the quotes.
	[attr~=value]	p[class~="special"]	Matches elements with an *attr* attribute whose value is exactly *value*, or contains *value* in its (space separated) list of values.
	[attr|=value]	div[lang|="zh"]	Matches elements with an *attr* attribute whose value is exactly *value* or begins with *value* immediately followed by a hyphen.
Substring Matching	[attr^=value]	li[class^="box-"]	Matches elements with an *attr* attribute (whose name is the value in square brackets), whose value begins with *value*.
	[attr$=value]	li[class$="-box"]	Matches elements with an *attr* attribute whose value ends with *value*.
	[attr*=value]	li[class*="box"]	Matches elements with an *attr* attribute whose value contains *value* anywhere within the string.

Read more : Link

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

The syntax of pseudo-classes:

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

- Although there are various CSS pseudo-classes, here we are going to discuss some of the most commonly used pseudo-classes. The widely used CSS classes are tabulated as follows:

All CSS Pseudo Classes

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
	Untitled	
	Untitled	
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element

Selector	Example	Example description
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

CSS 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

Syntax

The syntax of pseudo-elements:

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>

    <style>
      p::first-line {
        color: red;
        font-size: 40px;
      }

      p::before{
            content:"masai"
      }
    </style>
  </head>
  <body>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis
      impedit consequatur enim rerum exercitationem natus iusto, repellendus
      porro minus illo. Laborum, ducimus. Cum enim nisi assumenda voluptates
      recusandae. Ipsa, fugit. Vel repellendus quidem debitis repellat fugiat
      corporis blanditiis omnis perferendis harum, consequatur nemo commodi
      deserunt placeat iure delectus distinctio voluptatum? Aspernatur
      perspiciatis consequuntur alias magnam nihil dolores dolore hic illo.
      Illum itaque odit ipsam vitae officia saepe? Eveniet nobis quisquam porro
      officiis dolorum labore dolorem necessitatibus tempore incidunt
      repellendus ullam odio quis, molestias distinctio aliquid repudiandae
      officia voluptate recusandae accusamus!
    </p>
  </body>
</html>
```

All CSS Pseudo Elements

Selector	Example	Example description
::after	p::after	Insert content after every <p> element
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

Read more: Link