**ASIGNMENT**

**Difference Between div and span Tag in HTML**

Below is a table showcasing major differences between div and span in HTML.

|  |  |  |
| --- | --- | --- |
| **Feature** | **div** | **span** |
| **Type of Element** | Block-level | Inline |
| **Default Display** | Takes up the full width available, with a new line before and after | Only takes up as much width as necessary, without forcing new lines |
| **Primary Use** | Used to group larger blocks of content or other elements | Used to group a small chunk of HTML elements or to apply styling to part of a text |
| **Styling** | It can have width, height, margin, and padding, which affects the layout significantly. | Does not affect the layout with width and height; margin and padding are applied differently. |
| **Impact on Layout** | Significant, as it often creates a "box" for other elements | Minimal, typically used for styling text or small elements within text |
| **Nesting** | It can contain other block-level elements or inline elements | Typically contains only data or other inline elements |
| **Accessibility** | Not inherently accessible and requires additional attributes for accessibility. | Inherently inline but also requires additional attributes for accessibility when used for grouping |

|  |  |
| --- | --- |
| It accepts align attribute. | It does not accept align attribute. |
| This tag should be used to wrap a section, for highlighting that section. | This tag should be used to wrap any specific word that you want to highlight in your webpage. |

**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: content, padding, borders and margins. 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.

Example

Demonstration of the box model:

div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}

**Key components of the box model**

**1. Content Area**

* The content area is the central part of the CSS box model, containing the [main content](https://www.geeksforgeeks.org/html-main-tag/) (e.g., text, images, videos, or elements like <p> or <span>).
* It can be styled with [CSS properties](https://www.geeksforgeeks.org/css-properties-complete-reference/)like height and width.

The content edge refers to the four edges of the content area

* Left content edge
* Right content edge
* Top content edge
* Bottom content edge

**2. Padding Area**

* The padding area is the space between the content and the border of an element.
* It includes the areas highlighted in light green and skin color in the example.
* The distance between the [content edge](https://www.geeksforgeeks.org/what-does-meta-http-equivx-ua-compatible-contentieedge-do/)and the border is the padding.
* The border marks the end of the padding area.
* The padding area contributes to the element's total dimensions.
* Padding can be adjusted using [CSS properties](https://www.geeksforgeeks.org/css-properties-complete-reference/).
* It works similarly with box-sizing: content-box and box-sizing: border-box, but with slight calculation differences.

**3. Border Area**

* The area that marks the end of an element is called as the[border](https://www.geeksforgeeks.org/css-borders/)it is the outer fencing for the element.
* The default border properties are provided in CSS to control the thickness of this outer fencing.
* The border area also add 's up to the complete[height and width](https://www.geeksforgeeks.org/how-to-set-the-width-and-height-of-an-image-using-html/)of the element.
* The more the border [width](https://www.geeksforgeeks.org/css-width-property/) the more will be the height or width of the element.
* In the above image the area marked with skin color is called the border area.

**4. Margin Area**

* The area outside the border of an element is called the [margin area](https://www.geeksforgeeks.org/css-box-model/).
* Basically this area depends on the parent of the element.
* The distance between the border of the parent element and the border of the child element is called as the margin.
* CSS has provides certain [margin properties](https://www.geeksforgeeks.org/css-margins-padding/)to get control over this scenario.

**1. Content-Box(default property)**

When the user set's the value of the [box-sizing property](https://www.geeksforgeeks.org/css-box-sizing-property/)for an element as[content-box](https://www.geeksforgeeks.org/how-is-border-box-different-from-content-box/)or even if user do not set's it ,it remains by default as content-box and in the actual height and width of the element the dimensions of the content area as well as the padding area is added to constitute the final dimensions of the element.

**↔**​

<style>

div {

height: 200px;

width: 200px;

box-sizing: content-box;

padding-left: 20px;

padding-right: 20px;

border-left: 2px solid red;

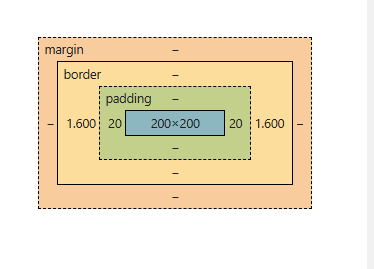
border-right: 2px solid red;

}

</style>

**↔**​

This code will create a box model with a[border line](https://www.geeksforgeeks.org/css-font-border/)width of 0.4px always and border-area of 1.6px and padding area as 20px width on both sides of the content area.



**Content Area (Width**) :The width of the content area is fixed at 200px.

**Padding**

* Padding adds extra space inside the element, around the content.
* Padding Left: 20px
* Padding Right: 20px
* Total padding width: 20px + 20px = 40px

**Border**

* The border, being solid, has a width, but it is calculated differently from the padding.
* Line Width of Border: 0.4px (the width of the line itself)
* Area of Border: 1.6px (the actual space the border occupies visually)
* Border width for both sides: 1.6px (left) + 1.6px (right) = 3.2px

**Total Width**

* Total width of the element can be calculated by adding the padding and border areas to the content area width.
* Formula for Total Width = (Padding-Left + Padding-Right + Border-Area-Left + Border-Area-Right) + Content Area Width
* Total Width = (20px + 20px + 1.6px + 1.6px) + 200px = 243.2px
* The total width of the element becomes 243.2px.

The reason the total width is increased unexpectedly is because[box-sizing: content-box](https://www.geeksforgeeks.org/css-box-sizing-property/)applies the width to the content area only .The padding and border are added outside the content area, leading to an increase in the overall [width and height](https://www.geeksforgeeks.org/html-width-height-attribute-vs-css-width-height-property/) of the element.

**2. Border-Box**

When the box-sizing property is set as [border-box](https://www.geeksforgeeks.org/what-is-the-use-of-box-sizing-property-in-css/) the actual dimensions of the element's remains same as that of the actual dimensions set by the user. The difference it makes is just that the size of the content area get's altered in a manner so that it could accommodate the[padding area](https://www.geeksforgeeks.org/css-padding/) and the[border area](https://www.geeksforgeeks.org/css-border-property/)so the resultant could be equal to the actual dimensions entered by the user.

**↔**​

<style>

div {

height: 200px;

width: 200px;

box-sizing:border-box;

padding-left: 20px;

padding-right: 20px;

border-left: 2px solid red;

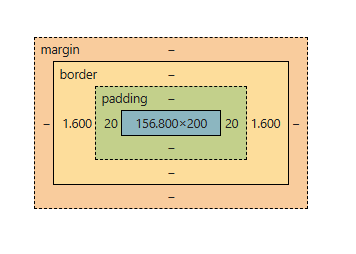
border-right: 2px solid red;

}

</style>

**↔**​

This code will create a box model by altering the dimensions specifically the width of the content area to accomodate the padding and the border area with the border line-width.



* **Width of Border and Padding Border width**: 0.4px (line width) and 1.6px + 1.6px = 3.2px (total border area).
* **Padding width**: 20px + 20px = 40px.
* **User-Entered Width** : The width entered by the user is 200px, which applies to the content area only when box-sizing: content-box is used.
* **Box-Sizing Behavior**:The box-sizing: content-box property adds the padding and border outside the content area, causing the total width to increase.
* **Adjusting Content Area Width**: To ensure the total width remains 200px, the extra width from padding and borders (40px + 3.2px = 43.2px) is subtracted from the total width.
* **New content area width** : 200px - 43.2px = 156.8px.
* **Final Width Calculation** : The final total width is: 156.8px (content area) + 40px (padding) + 3.2px (border) = 200px, ensuring the user’s entered width remains unchanged.

**Difference Between CSS Grid and Flexbox**

| **Property** | **Grid** | **Flexbox** |
| --- | --- | --- |
| Dimension | Two - Dimensional | One - Dimensional |
| Features | Can flex combination of items through space-occupying Features | Can push content element to extreme alignment |
| Support Type | Layout First | Content First |
| Primary Use Case | Creating complex layouts with rows and columns | Aligning items in a row or column |
| Performance | Can be less in performance due to very complex grids | Generally faster for simple layouts |

**CSS Specificity**

If there are two or more CSS rules that point to the same element, the selector with the highest specificity will "win", and its style declaration will be applied to that HTML element.

Think of specificity as a hierarchy that determines which style declaration is ultimately applied to an element.

Look at the following examples:

Example 1

Here, we have used the "p" element as selector, and specified a red color for this element. **Result:** The text will be red:

<html>  
<head>  
  <style>  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p>Hello World!</p>  
  
</body>  
</html>

Example 2

Here, we have added a class selector (named "test"), and specified a green color for this class. **Result:** The text will be green (even though we have specified a red color for the element selector "p"). This is because the class selector has higher priority:

<html>  
<head>  
  <style>  
    .test {color: green;}  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p class="test">Hello World!</p>  
  
</body>  
</html>

Specificity Hierarchy

Every CSS selector has a position in the specificity hierarchy.

|  |  |  |
| --- | --- | --- |
| Priority | Example | Description |
| Inline style | <h1 style="color: pink;"> | Highest priority, directly applied with the style attribute |
| Id selectors | #navbar | Second highest priority, identified by the unique id attribute of an element |
| Classes and pseudo-classes | .test, :hover | Third highest priority, targeted using class names |
| Attributes | [type="text"] | Low priority, applies to attributes |
| Elements and pseudo-elements | h1, ::before, ::after | Lowest priority, applies to HTML elements and pseudo-elements |