CSS

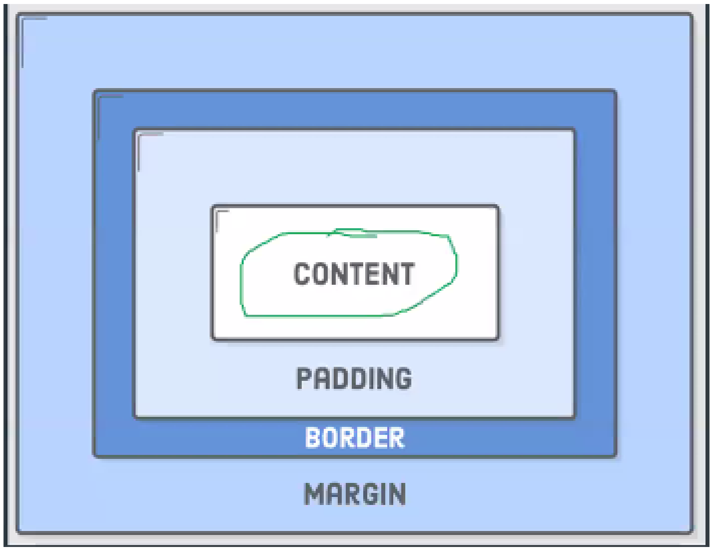
Topic:

- Box Model
- Margins, Padding & Border
- Position Attribute
- Display Properties

Box Model:

All the HTML elements follows the box model. The idea is every element can be represented in this box model. If we put multiple boxes together, that makes a HTML Web-page.

```
<p> content</p>
```



So, once the content is displayed is has padding, then border and finally the margin would be applied (default-style). For <h1> to <h6> certain margins are applied. But for other elements, none are provided; that is the default style.

For `<h1>this is my heading</h1>`, there is no padding or border, but only margin has been applied.

The body and <h1> both have margins which are over-lapping. The padding and content are together. The margin area is always outside the border.

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, if a particular color is given as background for content, the same would be taken by the padding as well.
- **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.

Important: When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.

Example

This <div> element will have a total width of 350px:

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Here is the calculation:

320px (width)

+ 20px (left + right padding)

+ 10px (left + right border)

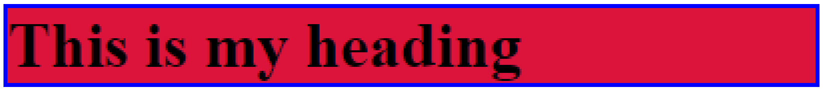
+ 0px (left + right margin)

= 350px

The total width of an element should be calculated like this:

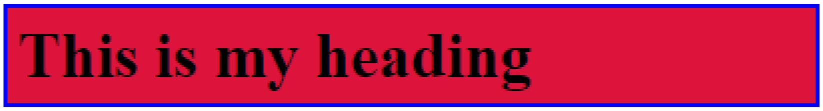
Total element width = **width** + **left** padding + **right** ***padding*** + **left** border + **right** ***border*** + **left** margin + **right** ***margin***

If you add border to the h1 element, the element starts exactly after the border i.e., to say, there is no space between the content area and the border. But if we provide padding, then there is space between the border and the padding area.



This is my heading

This is without padding



This is my heading

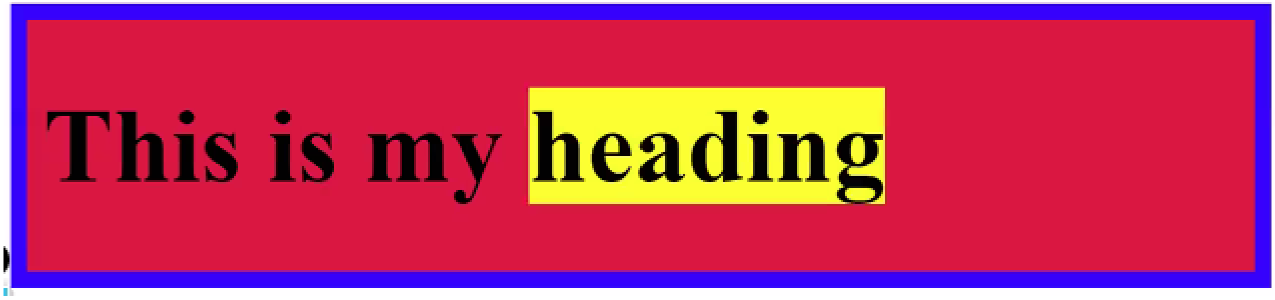
This is with padding. There is space between the blue border and the word 'This'.

All HTML elements include box models.

If we add span element inside the h1,

<h1>This is my <span>heading</span></h1>

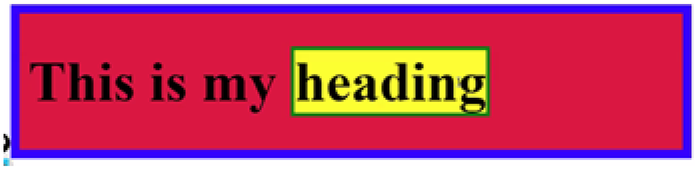
Span {background-color: yellow;}



This is my heading

Adding border to span element,

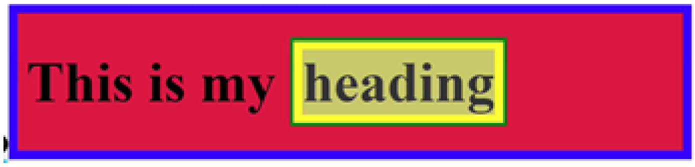
```
span  {
        background-color: yellow;
        border: 2px solid green;
}
```



This is my heading

Adding padding to the span element,

```
span  {
        background-color: yellow;
        border: 2px solid green;
        padding: 5px;
}
```



This is my heading

`<span>` being an inline element, the content size is shown as auto x auto. A block level element takes the complete width of the web page, so if we increase the padding area, the background color will also increase as the content and padding are tied together.

Margin:

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).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- **`margin-top`**
- **`margin-right`**
- **`margin-bottom`**
- **`margin-left`**

All the margin properties can have the following values:

- **`auto`** - the browser calculates the margin
- ***`length`*** - specifies a margin in px, pt, cm, etc.
- ***`%`*** - specifies a margin in % of the width of the containing element
- **`inherit`** - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed

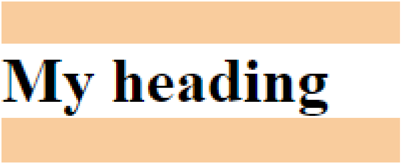
The **`auto`** Value

You can set the margin property to auto to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

In html

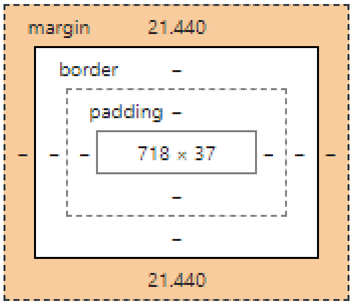
```
<h1>My heading</h1>
```



In css

```
body {
        padding: 0;
        margin:0;
}

h1      {
        margin-top: 12px;
}
```



Padding:

Padding is used to create space around an element’s content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

The padding property is a shorthand property for the following individual padding properties:

- padding-top
- padding-right
- padding-bottom
- padding-left

So, here is how it works:

If the padding property has four values (short hand):

```
div {
        padding: 10px 15px 20px 40px;
}
```

- **padding: 10px 15px 20px 40px;**
  - top padding is 10px
  - right padding is 15px
  - bottom padding is 20px
  - left padding is 40px

Property	***Description***
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

https://developer.mozilla.org/en-US/docs/Web/CSS/padding

Border:

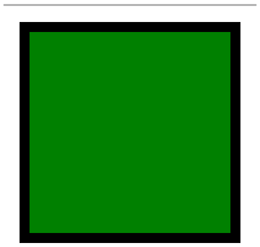
In html

```
        <div class="box"></div>
```

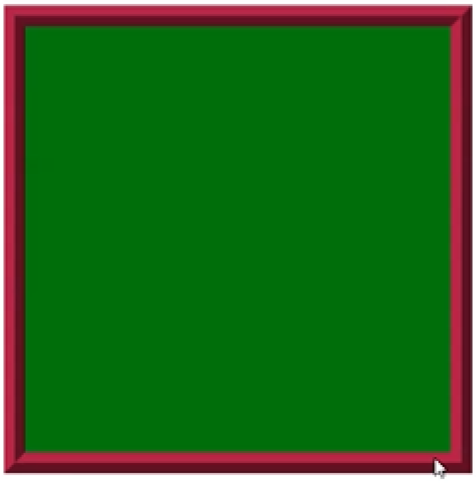
In css

```
<style type="text/css">
        .box {
                height: 100px;
                width: 100px;
                background-color: green;
                border: 5px solid #000000;
        }
</style>
```

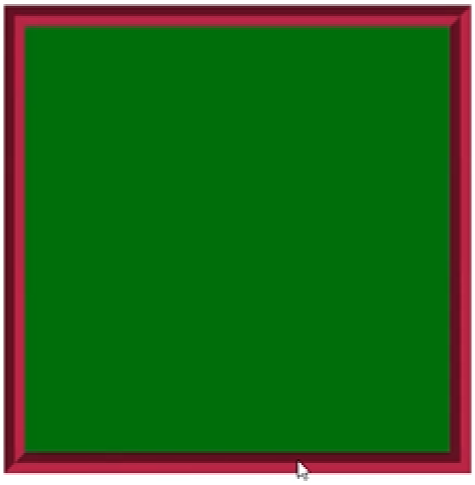
Output:



border-style: ridge;

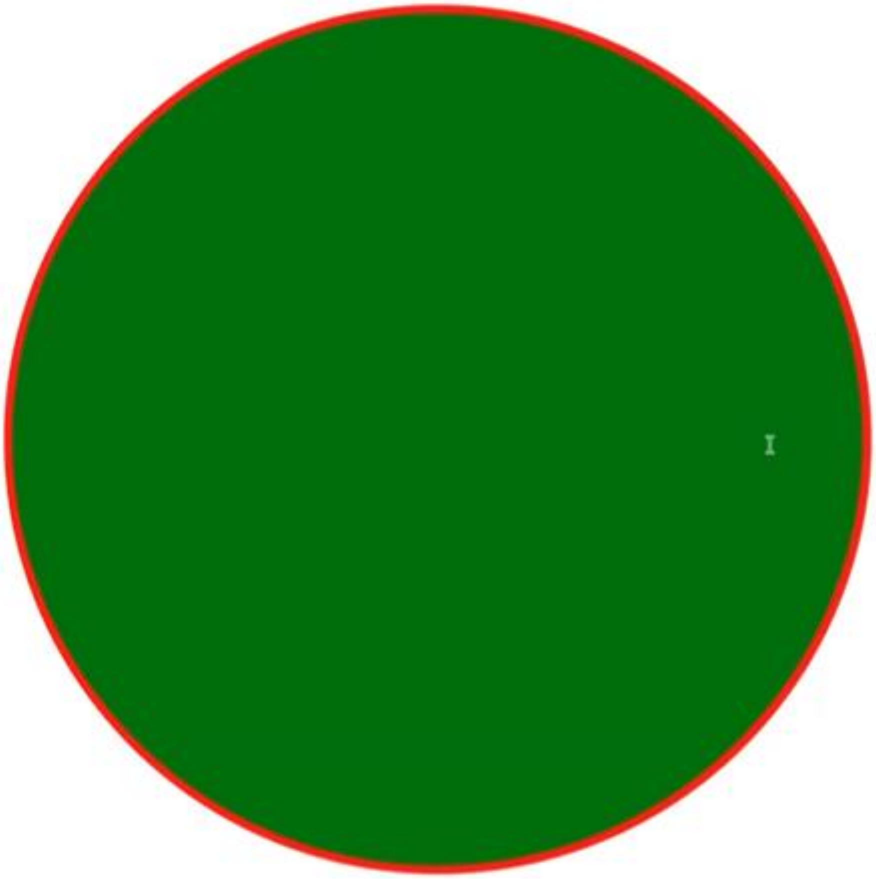


border-style: groove;



border: 2px solid red;

border-radius: 50%;



The border-style property specifies what kind of border to display. The following values are allowed:

`dotted` - Defines a dotted border

`dashed` - Defines a dashed border

`solid` - Defines a solid border

`double` - Defines a double border

`groove` - Defines a 3D grooved border. The effect depends on the border-color value

`ridge` - Defines a 3D ridged border. The effect depends on the border-color value

`inset` - Defines a 3D inset border. The effect depends on the border-color value

`outset` - Defines a 3D outset border. The effect depends on the border-color value

`none` - Defines no border

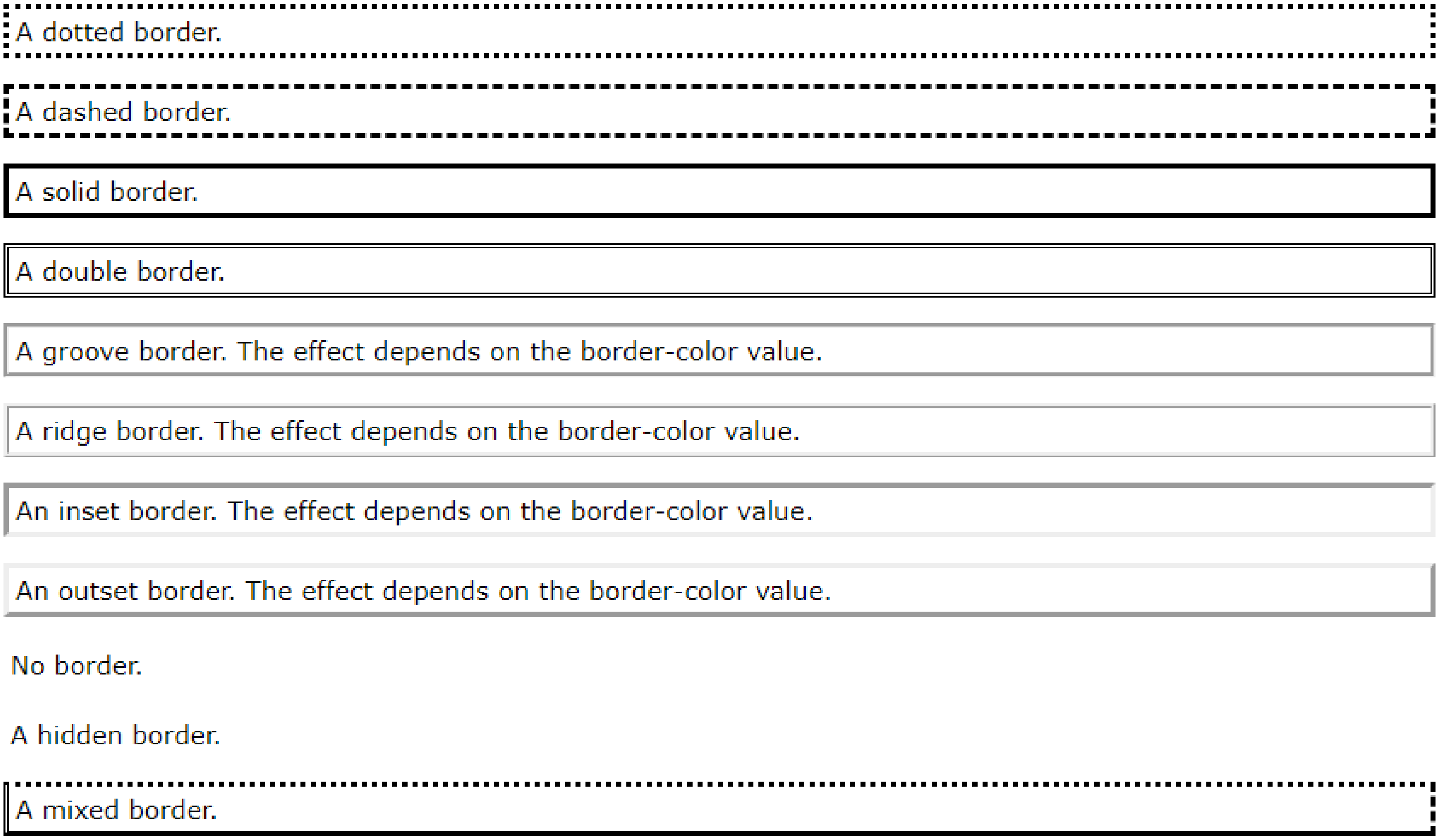
`hidden` - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Input:

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

Output:



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

All CSS Border Properties

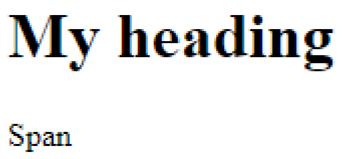
Property	***Description***
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-radius	Sets all the four border-*-radius properties for rounded corners
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders

https://developer.mozilla.org/en-US/docs/Web/CSS/border

Display Properties:

```
<h1>My heading</h1>

<span>Span</span>
```



My heading

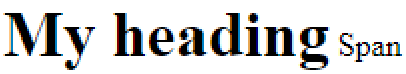
Span

The span element is displayed below the h1 and not next to the h1 element even though there is empty space next to it.

Now, in the css;

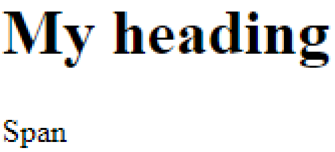
```
h1 {
        display: inline;
}
```

The output would now show the span element next to the h1 element, because the browser now takes h1 as inline element and not as block level element. Previously h1 (by default) was a block level element, but now, we have changed its display property from block level to inline, so the span element which is also an inline element would take the empty space next to the h1 element.



My heading Span

If you make span as block element, then the span would come down, because it is a block level it would take a new line and will not continue along with the h1 element.

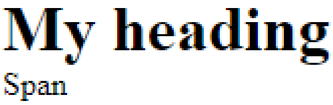


My heading

Span

When you have an inline element, the height and width would not apply to it.

```
        h1     {
                display: inline;
                height: 100px;
        }
        span {
                display: block;
        }
```

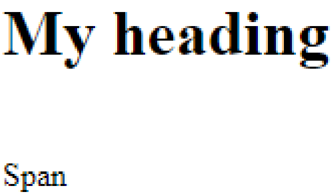


My heading

Span

Now, if you want the height and width to be specified for an inline element, then it should be displayed as *inline-block*, where you can specify height and width of an element.

```
        h1     {
                display: inline-block;
                height: 100px;
        }
        span {
                display: block;
        }
```



My heading

Span