ExcelShore®
Operational Excellence

CSS3

Agenda

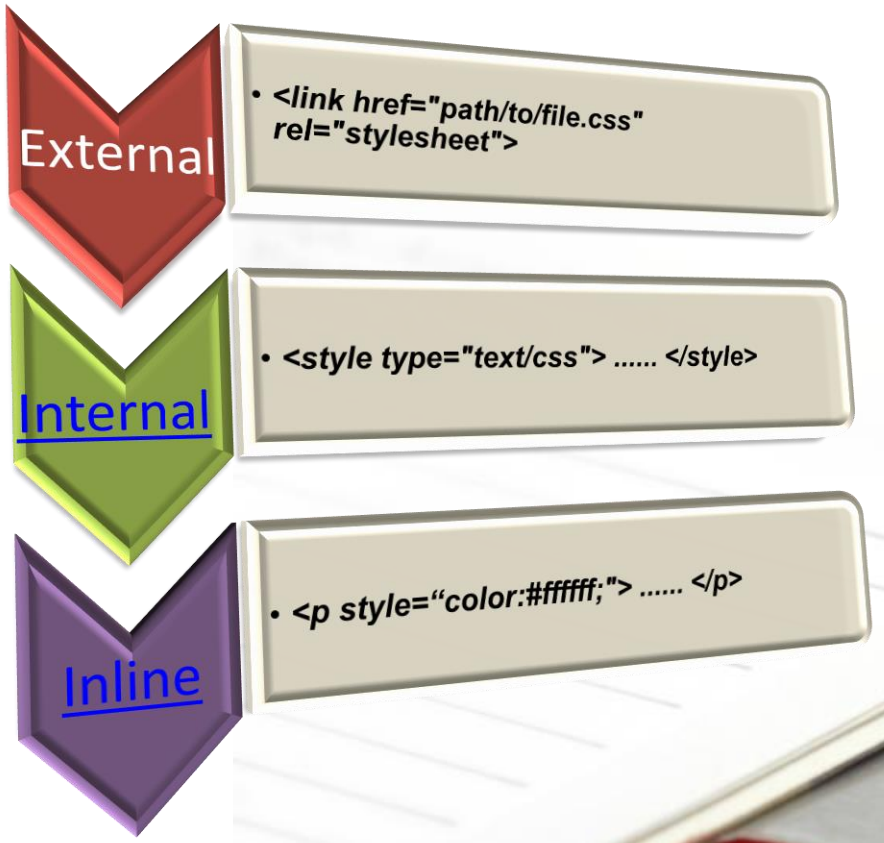
Topics
Introduction to Css
Types of Css
Css Declaration
Selectors in Css
Box Model
Priority schemes
Display
Inline , block , Inline-block , none
Visibility
hidden, visible

Topics
Position
Static, Relative, Fixed, Absolute
Float
Left, Right, Clear
Border
Box Shadow
Transformation
Multiple Columns

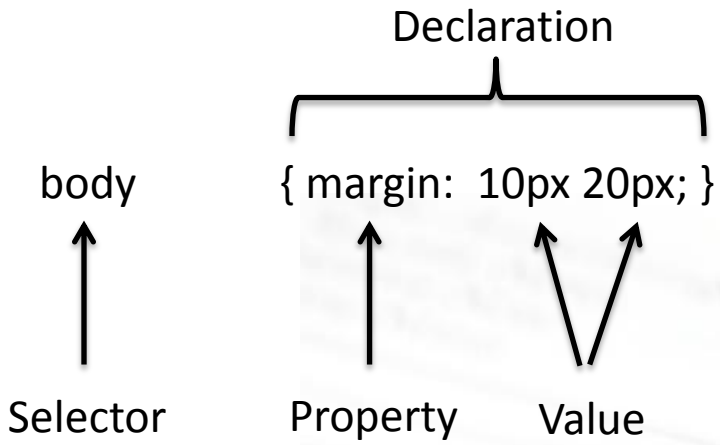
What / Why Css ?

- CSS stands for Cascading Style Sheets.
- Styles define how to display HTML elements.
- CSS is used to control the style and layout of Web pages.
- CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element.

Types of CSS



CSS Selectors



Selectors

- ID
- Class
- Element

CSS Selectors

body { margin: 10px 20px; } ⟵ Element

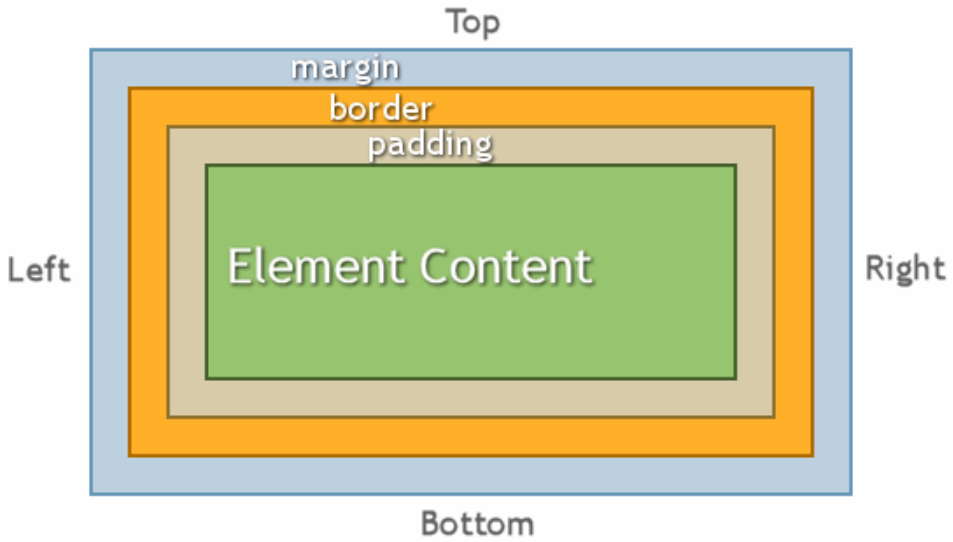
#slider-container { margin: 10px 20px 30px; } ⟵ ID

.blue-btn { color: blue; } ⟵ Class

Box Model

The CSS box model is essentially a box that wraps around HTML elements, and it consists of:

- Margins
- Borders
- Padding
- The actual content



Box Model

Margin:

Clears an area around the border. The margin does not have a background color, it is completely transparent.

Border:

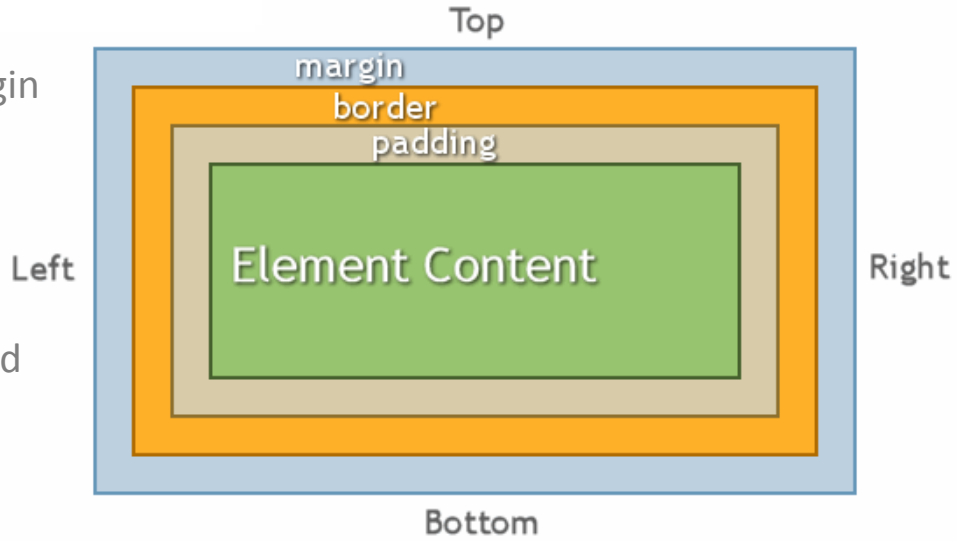
A border that goes around the padding and content. The border is inherited from the color property of the box.

Padding:

Clears an area around the content. The padding is affected by the background color of the box.

Content:

The content of the box, where text and images appear.



CSS – What's the difference

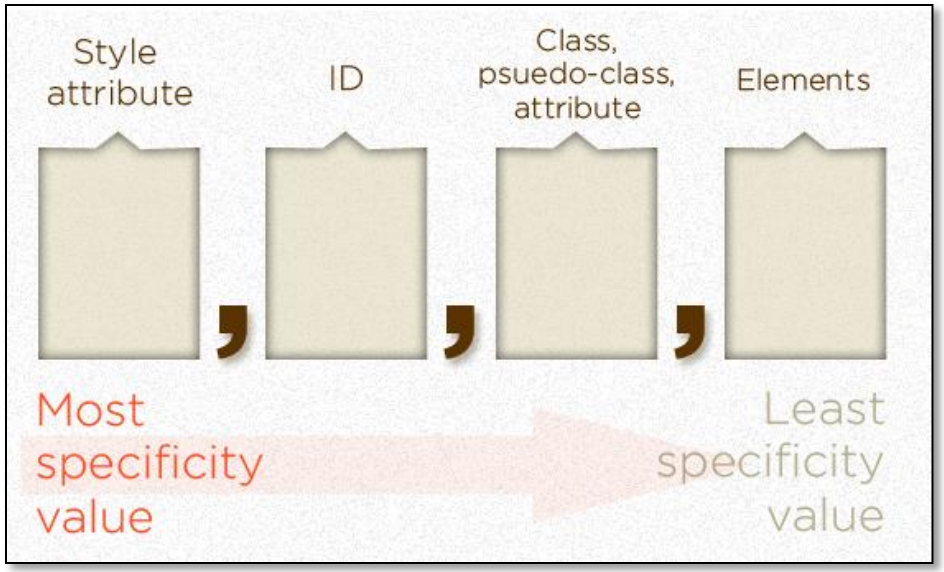
```
div                 { margin:  10px 20px; }

.blue-btn           { margin:  15px; }

#slider-container   { margin:  10px  20px  15px; }
```

```
<div id="slider-container" class="blue-btn"> … </div>
```

What if 2 same properties are assigned to one element ?

CSS Specificity

- Specificity is weight that is applied to a given css declaration.
- When specificity is equal to any of the multiple declarations, the last declaration found in the CSS is applied to the element.




```
*                /* a=0 b=0 c=0 -> specificity =   0 */
LI               /* a=0 b=0 c=1 -> specificity =   1 */
UL LI            /* a=0 b=0 c=2 -> specificity =   2 */
UL OL+LI         /* a=0 b=0 c=3 -> specificity =   3 */
H1 + *[REL=up]   /* a=0 b=1 c=1 -> specificity =  11 */
UL OL LI.red     /* a=0 b=1 c=3 -> specificity =  13 */
LI.red.level     /* a=0 b=2 c=1 -> specificity =  21 */
#x34y            /* a=1 b=0 c=0 -> specificity = 100 */
#s12:not(FOO)    /* a=1 b=0 c=1 -> specificity = 101 */
```

Display

Inline	Block	Inline-block
Respect left & right margins and padding, but not top & bottom.	Respect all of those – top, bottom, left and right margins & paddings.	Respect all of those – top, bottom, left and right margins & paddings.
Cannot have a width and height.	Force a line break after the element. Default width is 100%.	Respect height and width.
An inline element has no line break before or after it, and it tolerates HTML elements next to it.	A block element has some whitespace above and below it and does not tolerate any HTML elements next to it.	An inline-block element is placed as an inline element (on the same line as adjacent content), but it behaves as a block element.

Display: inline;

Imagine a element inside a .

If you give the element a height of 100px and a red border, it will look like this with.

display: inline

Ach, was muß man oft von bösen
Kindern hören oder lesen!
Wie zum Beispiel hier von diesen
Welche Max und Moritz hießen.
Die, anstatt durch weise Lehren
Sich zum Guten zu bekehren,
Oftmals noch darüber lachten
Und sich heimlich lustig machten.

Display : inline-block;

Imagine a element inside a .

If you give the element a height of 100px and a red border, it will look like this with.

display: inline-block

Ach, was muß man oft von bösen
Kindern hören oder lesen!
Wie zum Beispiel hier von diesen
Welche Max und Moritz hießen.

Die, anstatt durch weise Lehren
Sich zum Guten zu bekehren,
Oftmals noch darüber lachten
Und sich heimlich lustig machten.

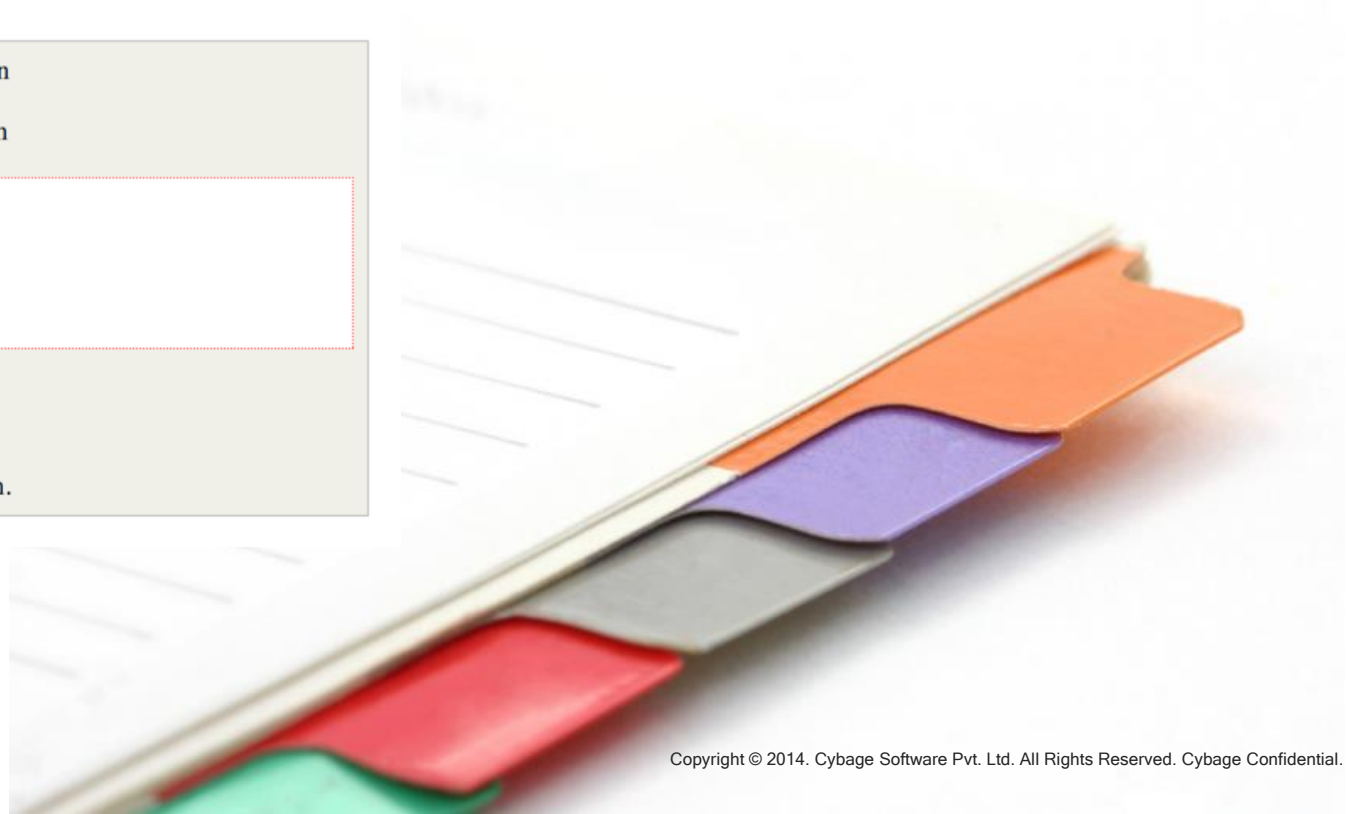
Display : block;

Imagine a element inside a .

If you give the element a height of 100px and a red border, it will look like this with.

display: block

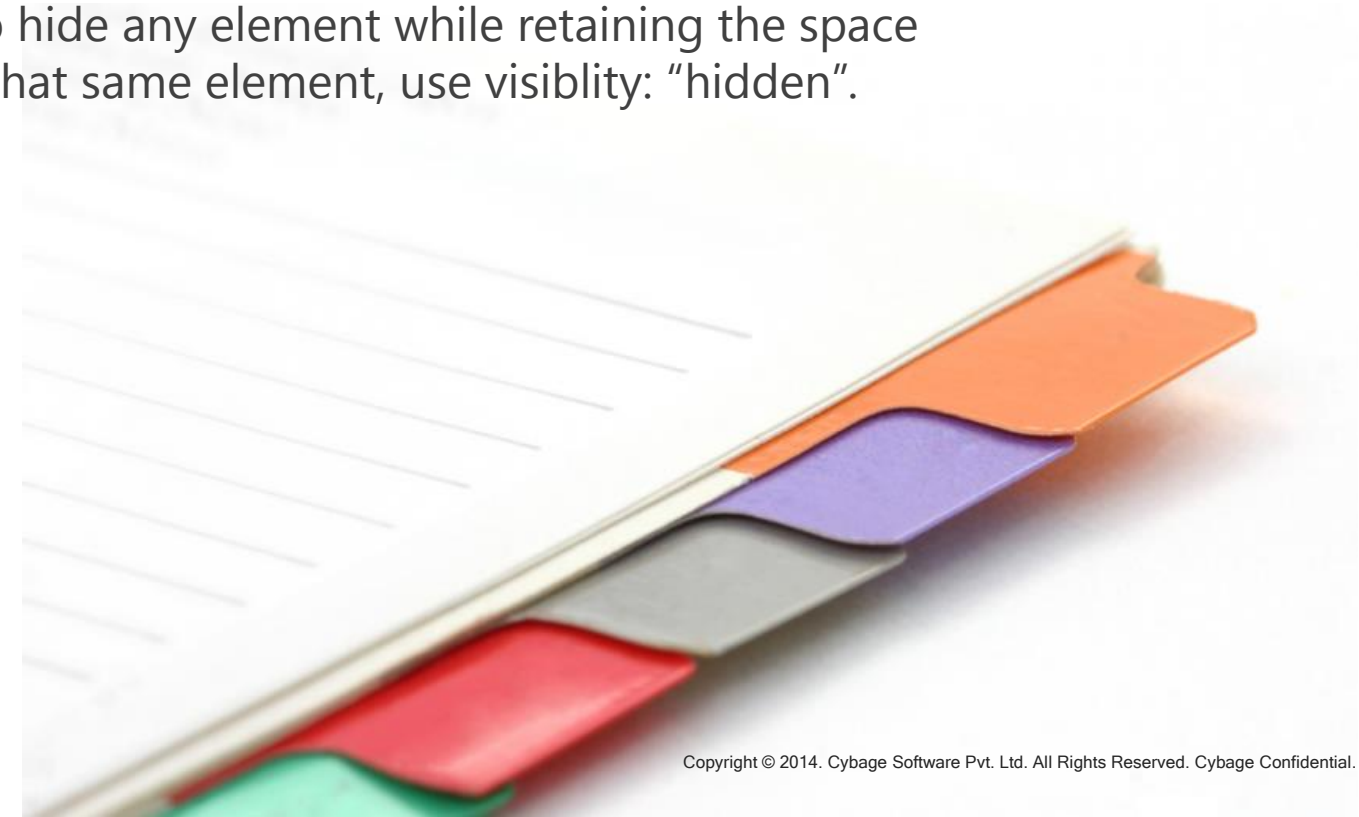
Ach, was muß man oft von bösen
Kindern hören oder lesen!
Wie zum Beispiel hier von diesen
Welche
Max und Moritz
hießen.
Die, anstatt durch weise Lehren
Sich zum Guten zu bekehren,
Oftmals noch darüber lachten
Und sich heimlich lustig machten.



Visibility

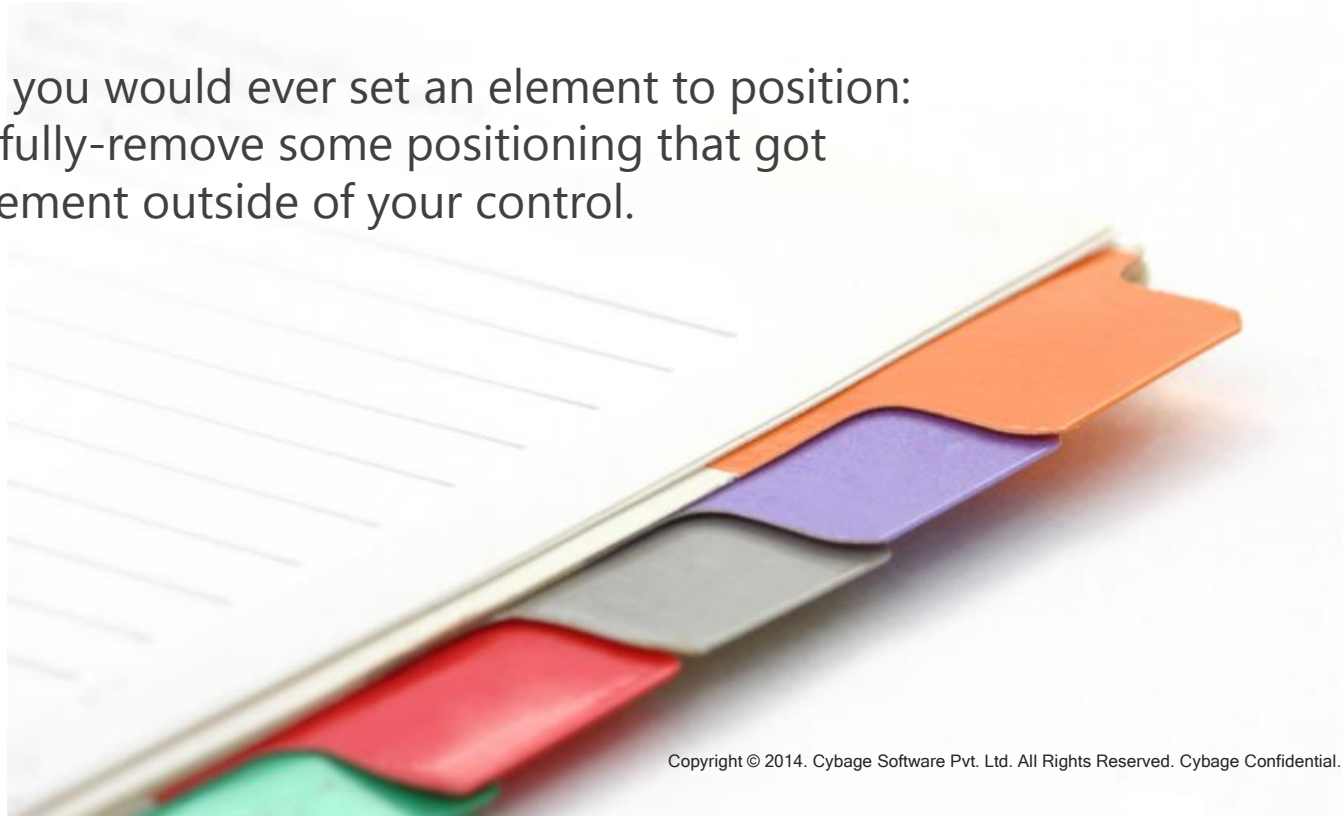
- By default the visibility of any element is set to “visible”

- If you want to hide any element while retaining the space occupied by that same element, use visiblity: “hidden”.



Position: static;

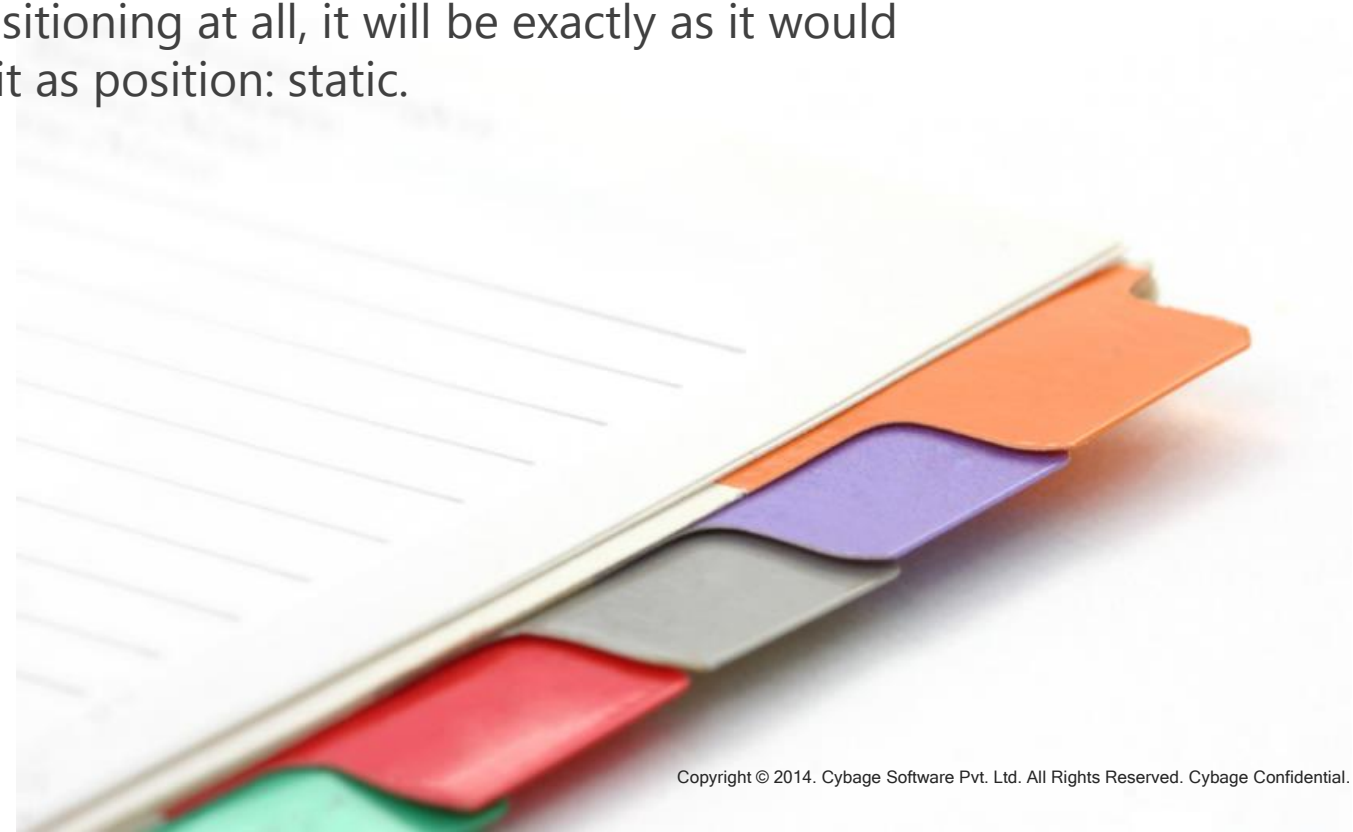
- This is the default for every single page element.
- Static doesn't mean much, it just means that the element will flow into the page as it normally would.
- The only reason you would ever set an element to position: static is to forcefully-remove some positioning that got applied to an element outside of your control.



Position: relative;

- What it really means is "relative to itself".

- If you set position: relative; on an element but no other positioning attributes (top, left, bottom or right), it will no effect on it's positioning at all, it will be exactly as it would be if you leave it as position: static.



Position: relative;

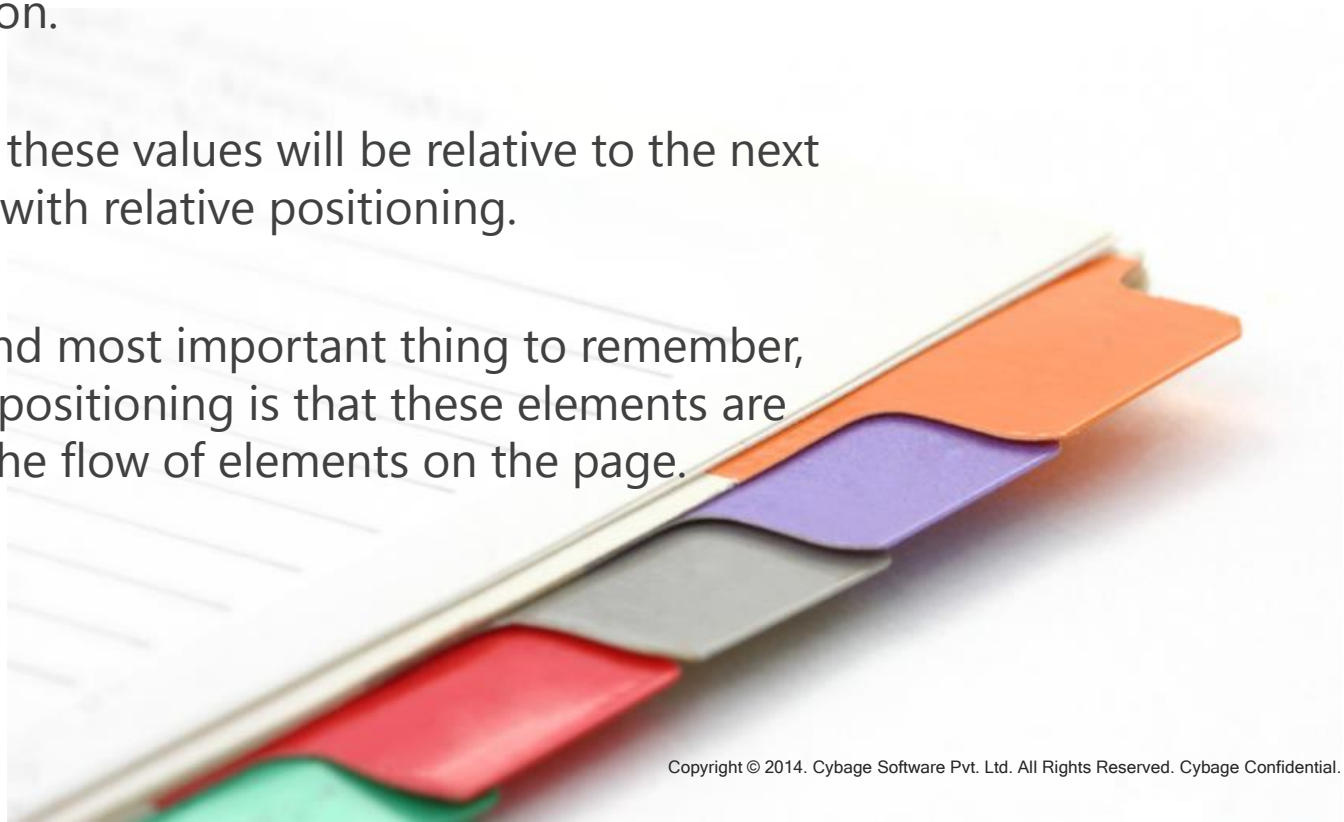
There are two other things that you should be aware of when you set position: relative on an element

- It introduces z-index on that element, which means that the element will now appear on **top** of any other **statically** positioned element.
- It limits the **scope** of **absolutely** positioned child elements.

Position: absolute;

This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it.

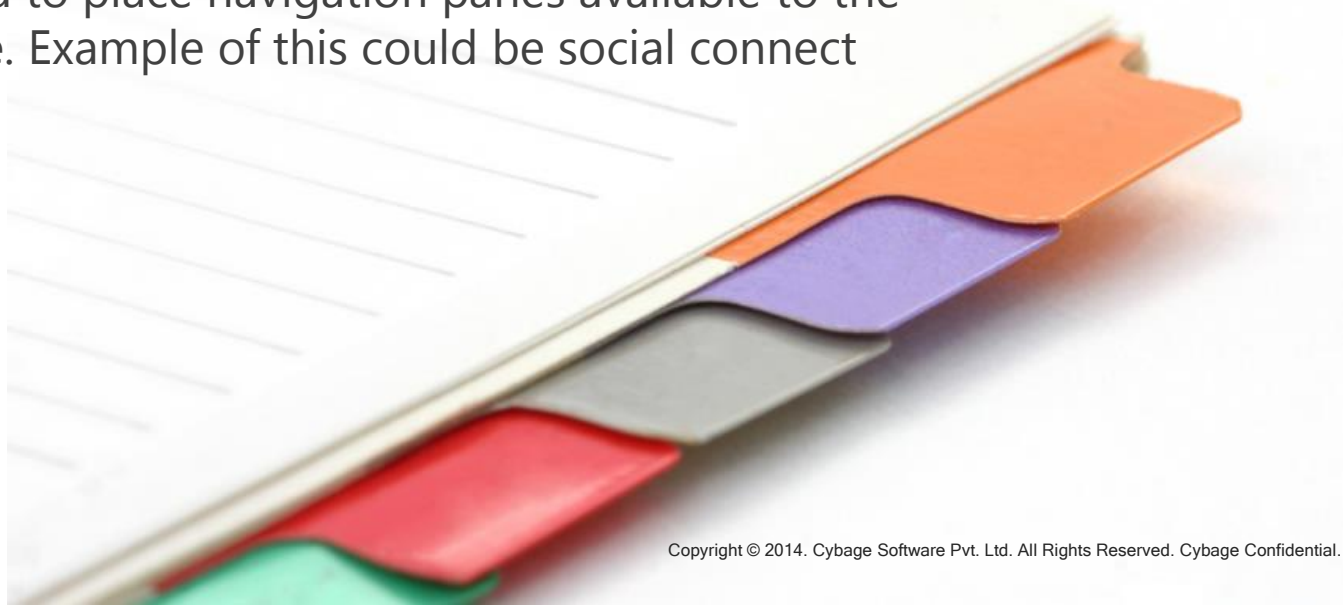
- You use the positioning attributes top, left bottom and right to set the location.
- Remember that these values will be relative to the next parent element with relative positioning.
- The trade-off, and most important thing to remember, about absolute positioning is that these elements are removed from the flow of elements on the page.



Position: fixed;

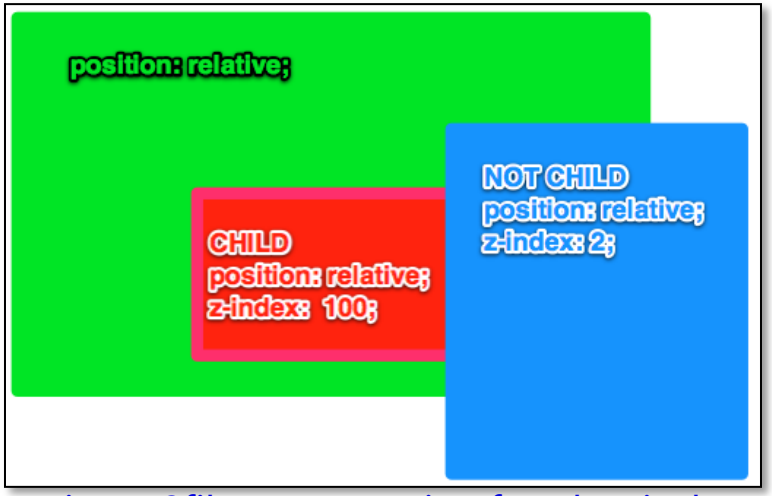
A fixed position element is positioned relative to the *viewport*, or the browser window itself.

- The viewport doesn't change when the window is scrolled, so a fixed positioned element will stay right where it is when the page is scrolled.
- It is usually used to place navigation panes available to the user all the time. Example of this could be social connect icon / header.



CSS Z-index

- The z-index property specifies the stack order of an element.
- An element with greater stack order is always in front of an element with a lower stack order.

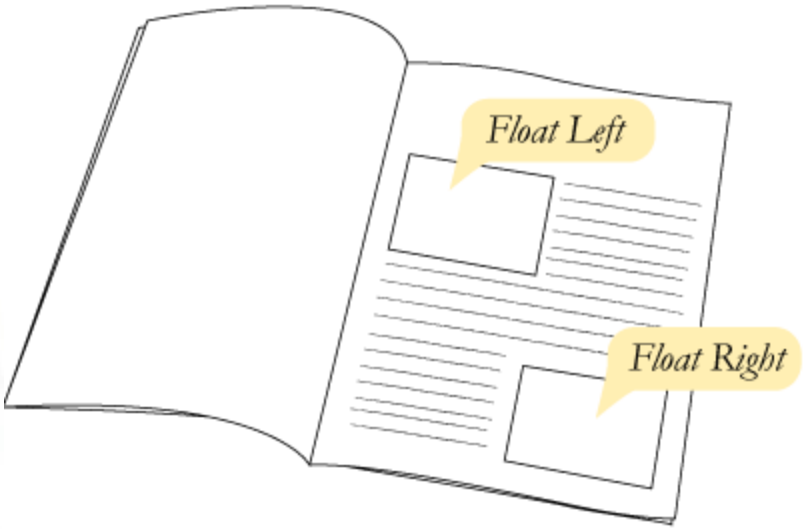


Reference link: http://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_style_zindex

Note: z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).

Float

- With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

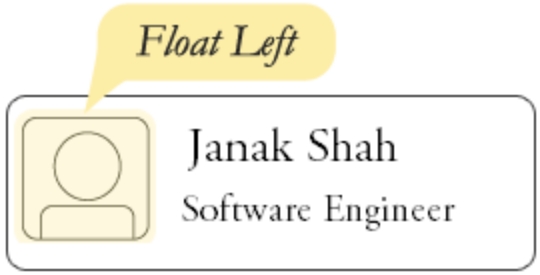


- Float is often used with images, but it is also useful when working with layouts.

- A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

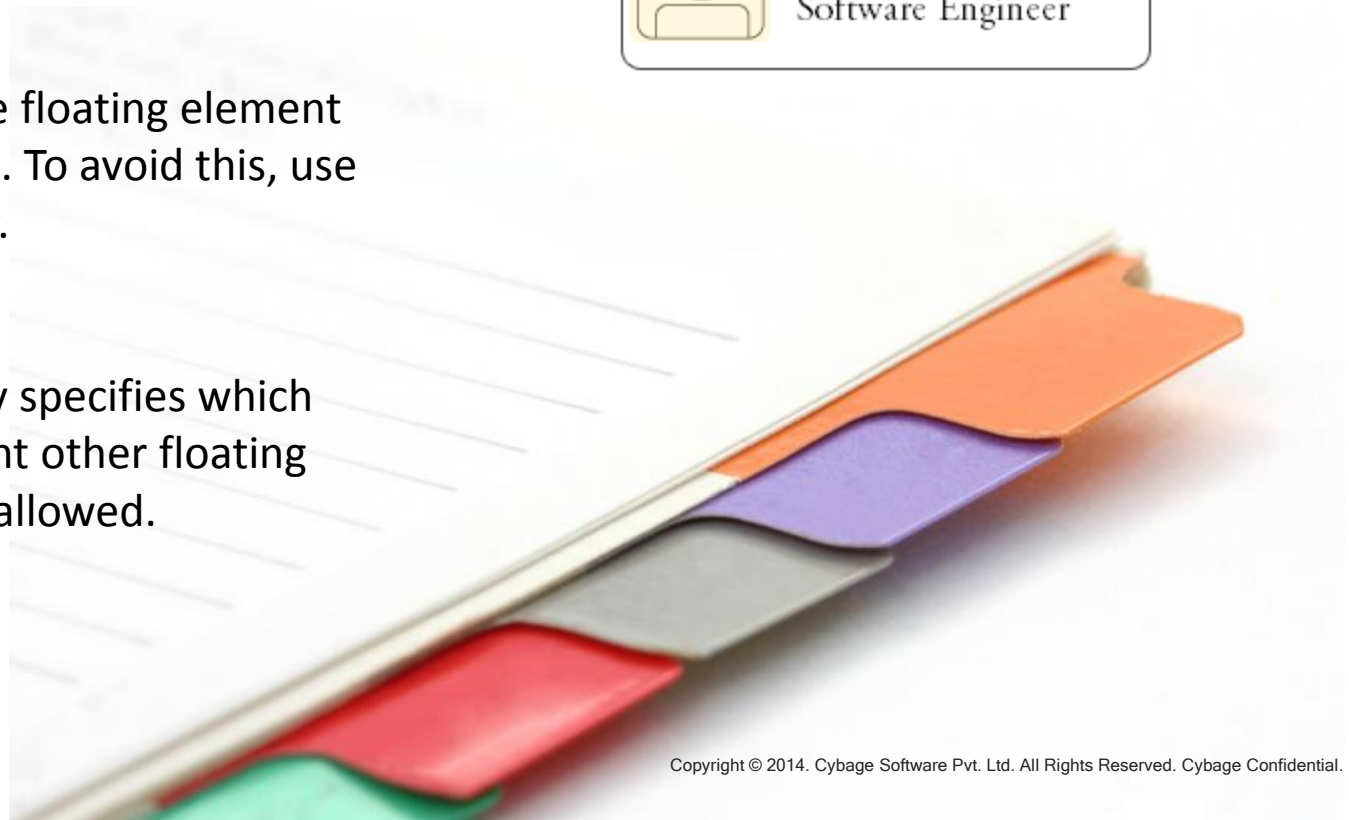
Float

- If you place several floating elements after each other, they will float next to each other if there is any room for it.



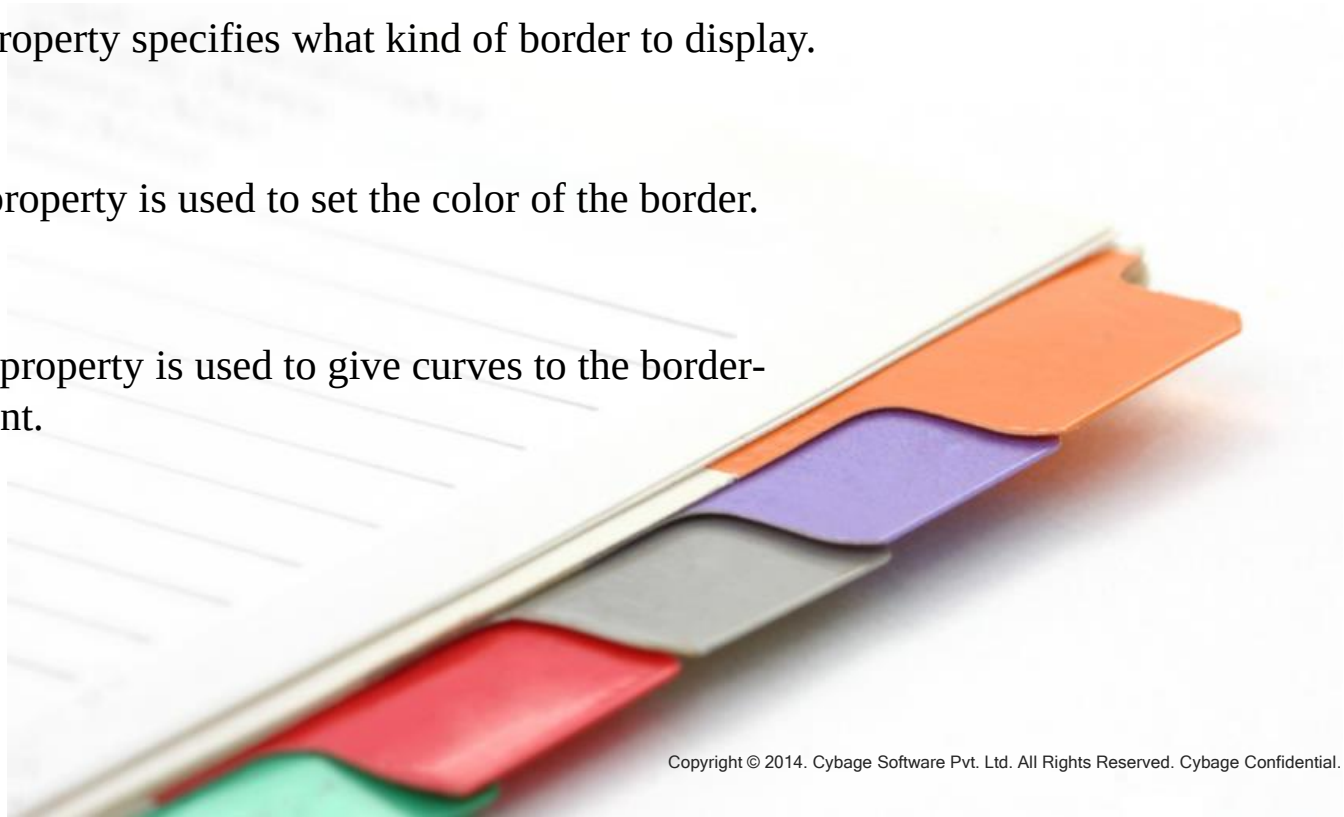
- Elements after the floating element will flow around it. To avoid this, use the clear property.

- The clear property specifies which sides of an element other floating elements are not allowed.



border

- border-width
  The border-width property is used to set the width of the border.
- border-style
  The border-style property specifies what kind of border to display.
- border-color
  The border-color property is used to set the color of the border.
- border-radius
  The border-radius property is used to give curves to the border-edges of the element.



border-style

- border-style

  The border-style property specifies what kind of border to display.

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

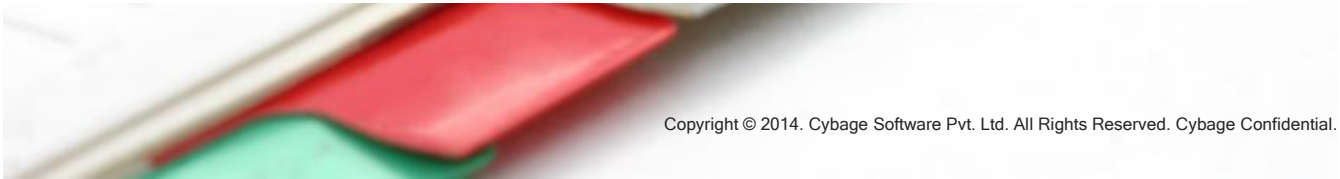
double: Defines two borders. The width of the two borders are the same as the border-width value

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



border-color

- border-color

  The border-color property is used to set the color of the border.

The color can be set by:

- name – **border-color: red;**
- RGB – **border-color: rgb(255,0,0);**
- Hex – **border-color: #ff0000;**

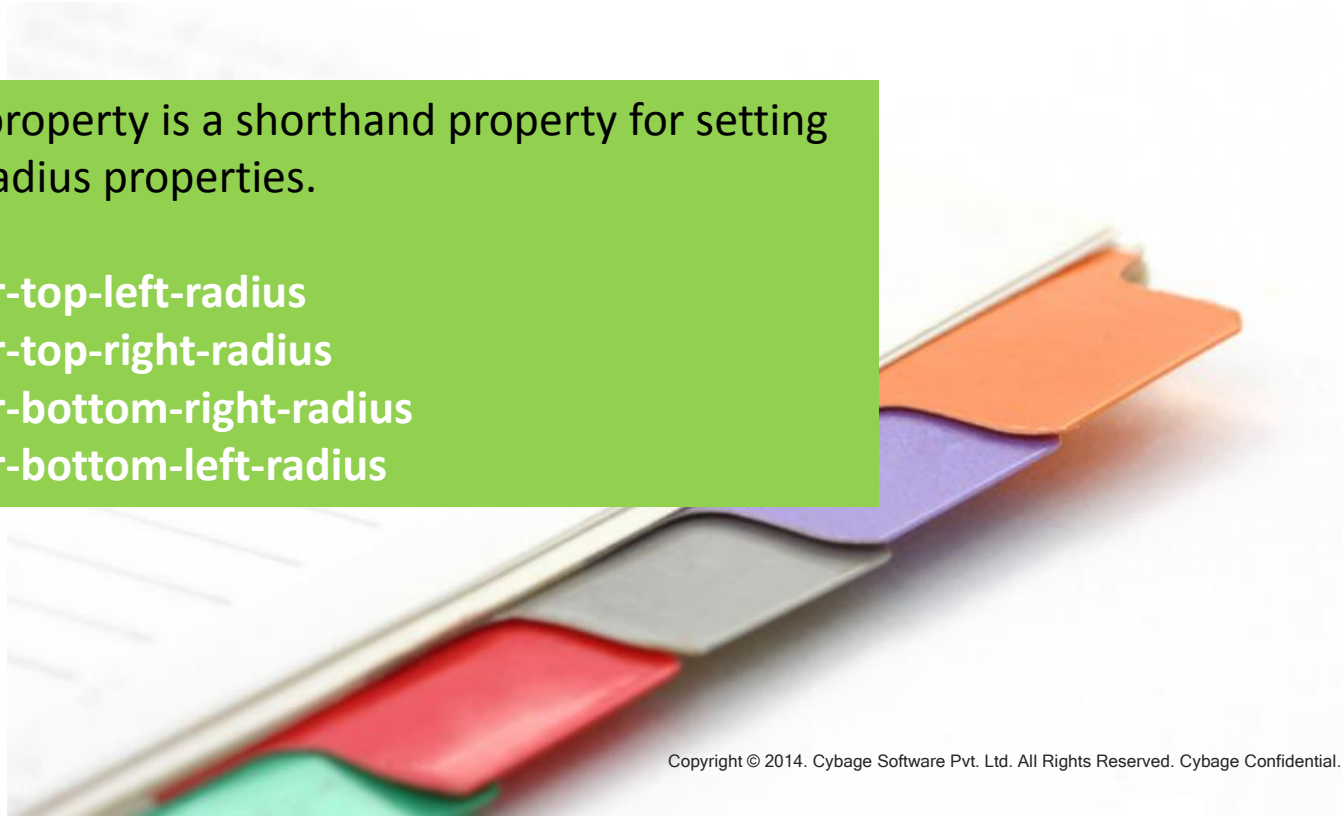
Border-radius

- border-radius

  The border-radius property is used to give curves to the border-edges of the element.

The border-radius property is a shorthand property for setting the four border-*-radius properties.

- **border-top-left-radius**
- **border-top-right-radius**
- **border-bottom-right-radius**
- **border-bottom-left-radius**



border

So, to have a border to a div, we need to specify.

- border-width: 1px;
- border-style: dotted;
- border-color: blue;
- border-top-left-radius: 2px;
- border-top-right-radius : 2px;
- border-bottom-right-radius : 2px;
- border-bottom-left-radius : 2px;

Isn’t it tedious to write so much of text just to have one small border ???

Border - shorthand

- ~~border-width: 1px;~~
- ~~border-style: dotted;~~
- ~~border-color: blue;~~

Border: 1px dotted blue;

- ~~border-top-left-radius: 2px;~~
- ~~border-top-right-radius : 2px;~~
- ~~border-bottom-right-radius : 2px;~~
- ~~border-bottom-left-radius : 2px;~~

Border-radius: 2px;

https://www.sitepoint.com/introduction-css-shorthand/

box-shadow

You can give shadows to your elements to give various effects

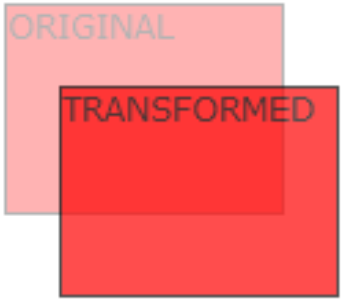
box-shadow: [shadow-type] | x-offset | y-offset | blur | blur-offset | color

shadow-type	To specify the type of shadow.
x-offset	To position the shadow along the x-axis.
y-offset	To position the shadow along the y-axis.
blur	To set the amount of blur.
blur-offset	To set the reach of blur.
color	To set the color of the shadow.

Transformation

Translate

transform: translate(50px,100px);



Rotate

transform: rotate(30deg);

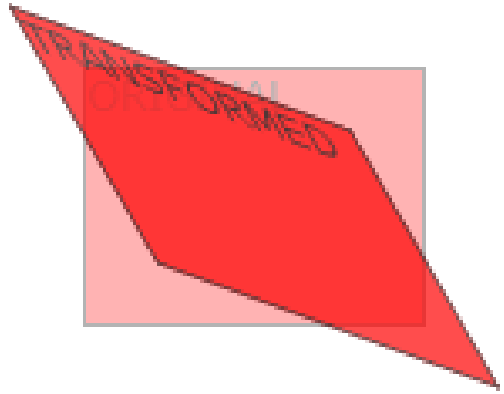
Scale

transform: scale(2,4);

Transformation

Skew

transform: skew(30deg,20deg);

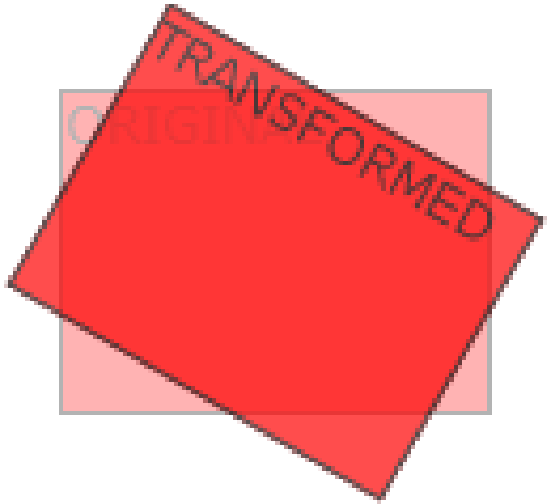


Matrix

transform:

matrix(0.866, 0.5, -0.5, 0.866 ,0,0);

(rotate, scale, translate, skew)



Transition

Transitions allow the values of CSS properties to change over time, essentially providing simple animations.
For example, if a link changes color on hover, you can have it gradually fade from one color to the other, instead of a sudden change

Here are the steps to create a simple transition using only CSS:

- Declare the original state of the element in the default style declaration.
- Declare the final state of your transitioned element; for example, in a hover state.
- Include the transition functions in your default style declaration, using a few different properties: transition-property, transition-duration,transition-timing-function, and transition-delay.

Transition

list of properties that can be transitioned:			
background-color	Left	Letter-spacing	Text-indent
background-position	Right	Vertical-align	Text-shadow
border-color	color	Visibility	Line-height
border-spacing	Font-size	Word-spacing	Margin
border-width	Height	Z-index	padding
Top	width	bottom	Opacity

Transition-duration

The transition-duration property sets how long the transition will take. You can specify this either in seconds (s) or milliseconds (ms). We'd like our animation to be fairly quick, so we'll specify 0.2 seconds, or 200 milliseconds:

```
-webkit-transition-duration: 0.2s;
-moz-transition-duration: 0.2s;
-o-transition-duration: 0.2s;
transition-duration: 0.2s;
```

Transition-timing-function

The transition-timing-function lets you control the pace of the transition in even more granular detail. Do you want your animation to start off slow and get faster, start off fast and end slow.
You can specify one of the key terms **ease, linear, ease-in, ease-out, or easein-out.** The best way to familiarize yourself with them is to play around and try them all.

```
-webkit-transition-timing-function: ease-out;
-moz-transition-timing-function: ease-out;
-o-transition-timing-function: ease-out;
transition-timing-function: ease-out;
```

Transition-delay

Finally, by using the transition-delay property, it's also possible to introduce a delay before the animation begins. Normally, a transition begins immediately, so the default is 0. Include the number of milliseconds (ms) or seconds (s) to delay the transition:

```
-webkit-transition-delay: 250ms;
-moz-transition-delay: 250ms;
-o-transition-delay: 250ms;
transition-delay: 250ms;
```

Transition-shorthand

With four transition properties and three vendor prefixes, you could wind up with 16 lines of CSS for a single transition.
The transition property is shorthand for the four transition functions described above.

Transition-original property

```
#ad2 h1 span {
  -webkit-transition-property: -webkit-transform, color;
  -moz-transition-property: -moz-transform, color;
  -o-transition-property: -o-transform, color;
  transition-property: transform, color;
  -webkit-transition-duration: 0.2s;
  -moz-transition-duration: 0.2s;
  -o-transition-duration: 0.2s;
  transition-duration: 0.2s;
  -webkit-transition-timing-function: ease-out;
  -moz-transition-timing-function: ease-out;
  -o-transition-timing-function: ease-out;
  transition-timing-function: ease-out;
}
```

Transition-shorthand property

```
#ad2 h1 span {
  -webkit-transition: -webkit-transform 0.2s ease-out;
  -moz-transition: -moz-transform 0.2s ease-out;
  -o-transition: -o-transform 0.2s ease-out;
  transition: transform 0.2s ease-out;
}
```

Transition-shorthand-rule

Note that order of the values is important and must be as follows (though you don’t need to specify all four values):

1. transition-property
2. transition-duration
3. transition-function
4. transition-delay

Multiple columns

With CSS3 columns, the browser determines when to end one column and begin the next without requiring any extra markup

1. column-width
   The column-width property is like having a min-width for your columns. The browser will include as many columns of at least the given width as it can to fill up the element

2. column-gap
   The column-gap property specifies the width of the space between columns.

3. column-count
   The column-count property specifies the number of columns desired, and the maximum number of columns allowed. The default value of auto means that the element has one column.

Multiple columns

With CSS3 columns, the browser determines when to end one column and begin the next without requiring any extra markup

```
#articlex {
    -webkit-column-width: 150px;
    -moz-column-width: 150px;
    column-width: 150px;
}
```

```
#articlex {
    -webkit-column-count: 3;
    -moz-column-count: 3;
    column-count: 3;
}
```

```
#articlex {
    -webkit-column-gap: 10px;
    -moz-column-gap: 10px;
    column-gap: 10px;
}
```

ExcelShore®
Operational Excellence

Thank You!