Introduction to CSS3: Part.2

Introduction to Internet and Web

부산대학교 정보·의생명 공학대학
정보컴퓨터공학부

Table of Contents

CSS UNITS

CSS Units

- CSS has several different units for expressing a length.
- Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.
- Length is a number followed by a length unit, such as 10px, 2em, etc.
- A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 60px;
}
p {
  font-size: 25px;
  line-height: 50px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>

```

Absolute Lengths

- The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

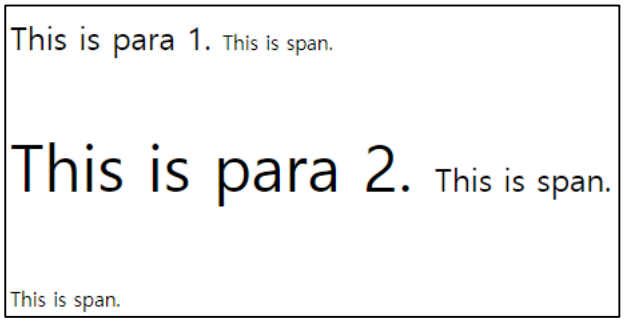
- Absolute length units are not recommended for use on screen, because screen sizes vary so much.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

```
<!DOCTYPE html>
<html>
<head>
<style>
#first{ font-size: 9.6px;}
#second{ font-size: 0.1in;}
#third{ font-size: 0.254cm;}
</style>
</head>
<body>
<p id="first">This is a
paragraph.</p>
<p id="second">This is a
paragraph.</p>
<p id="third">This is a
paragraph.</p>
</body>
</html>
```

Relative Lengths

- **Relative length units specify a length relative to another length property.**
  - Relative length units scales better between different rendering mediums.
- ***em***
  - Relative to the font-size of the element (2em means 2 times the size of the current font)
  - the default size for normal text, like paragraphs, is 16px (16px=1em).
- ***rem***
  - Relative to font-size of the root element
- ***%***
  - Relative to the parent element




```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  p.one { font-size: 16px;}
6  p.two { font-size: 64px;}
7  span { font-size: 0.5em;}
8  </style>
9  </head>
10 <body>
11 <p class="one">This is para 1.
12     <span>This is span.</span>
13 </p>
14 <p class="two">This is para 2.
15     <span>This is span.</span>
16 </p>
17 <span>This is span.</span>
18
19 </body>
20 </html>
21
22
23
```

CSS DISPLAY PROPERTY

Block-level Elements / Inline Elements

❖ The display property specifies if/how an element is displayed

- block, inline, flex, grid, ...

❖ Block-level Elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
- The element is a block-level element.
- Examples of block-level elements:
  - , <h1> - <h6>, <p>, <form>, <header>, <footer>, <section>

❖ Inline Elements

- An inline element does not start on a new line and only takes up as much width as necessary.
- This is an inline element inside a paragraph.
- Examples of inline elements:
  - , <a>, <img>

Override The Default Display Value

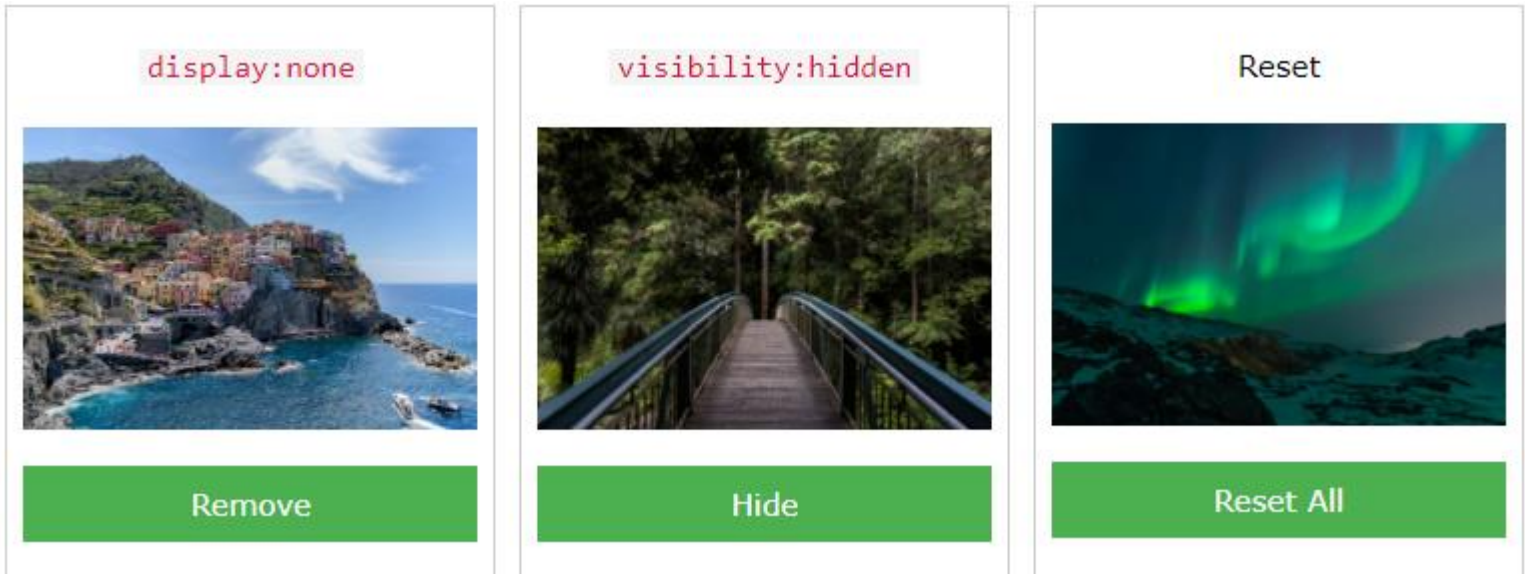
- Changing an inline element to a block element, or vice versa.

- A common example is making inline <li> elements for horizontal menus:

```
<!DOCTYPE html>
<html>
<head>
<style>
li { display: inline; }
</style>
</head>
<body>
<ul>
  <li><a href="/html/default.asp"
target="_blank">HTML</a></li>
  <li><a href="/css/default.asp"
target="_blank">CSS</a></li>
  <li><a href="/js/default.asp"
target="_blank">JavaScript</a></l
i>
</ul>
</body>
</html>
```

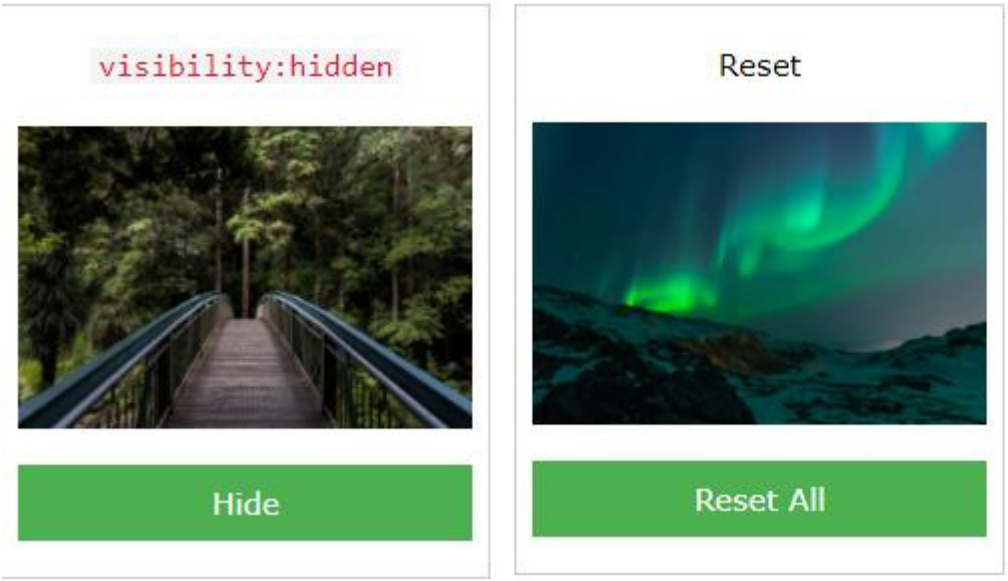
Hide an Element - display:none or visibility:hidden?

❖ Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:



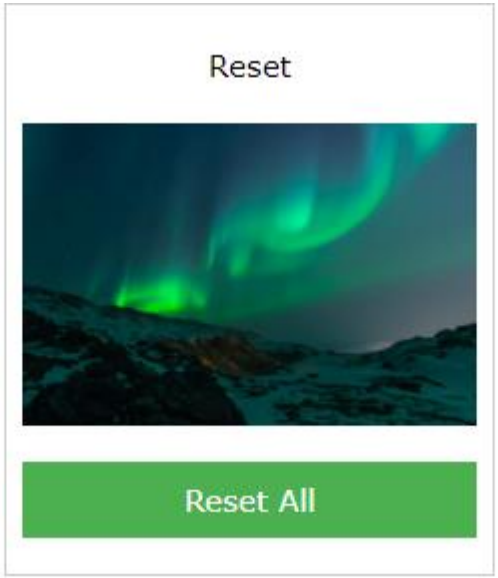
Hide an Element - display:none or visibility:hidden?

❖ Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:



Hide an Element - display:none or visibility:hidden?

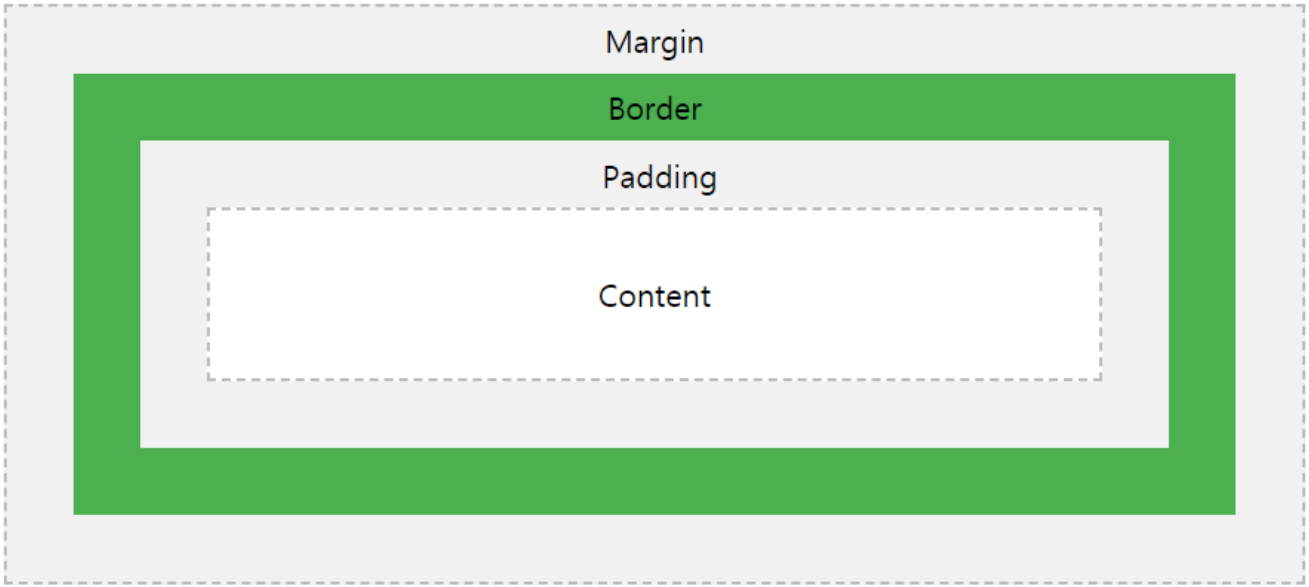
- visibility:hidden; also hides an element.
- However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:



CSS BOX MODEL

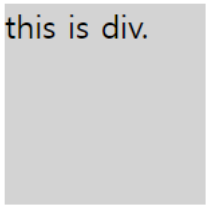
The CSS Box Mode

❖ **All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.**



- **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

width and height




```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  div {
6    background-color: lightgrey;
7    width: 100px;
8    height: 100px;
9  }
10 </style>
11 </head>
12 <body>
13
14 <div>this is div.</div>
15
16 </body>
17 </html>
18
19
20
21
22
23
```

Border

❖ border property

- border: [width] [style] [color];
  - a shorthand property for border-width, border-style, border-color properties.
  - e.g., border-width: 20px; border-style: solid; border-color: black;

❖ border width

- border-width: [top/bottom/right/left]
- border-width: [top/bottom] [right/left]
- border-width: [top] [right/left] [bottom]
- border-width: [top] [right] [bottom] [left]

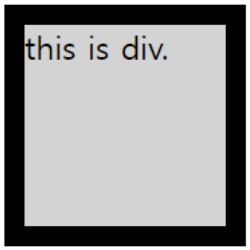
❖ border-style

- solid, dotted, dashed, ...
- border-style: [top] [right] [left] [bottom]

❖ border-color

- border-color: [top] [right] [left] [bottom]

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  div {
6    background-color: lightgrey;
7    width: 100px;
8    height: 100px;
9    border: 10px solid black;
10 }
11 </style>
12 </head>
13 <body>
14
15 <div>this is div.</div>
16
17 </body>
18 </html>
19
20
21
22
23
```



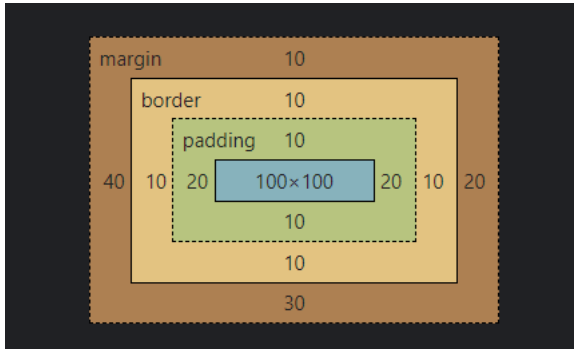
Margin and Padding

❖ margin

- margin [top] [right] [left] [bottom]
- margin-top
- margin-right
- margin-bottom
- margin-left

❖ padding

- padding [top] [right] [left] [bottom]
- padding-top
- padding-right
- padding-bottom
- padding-left




```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  div {
6    background-color: lightgrey;
7    width: 100px;
8    height: 100px;
9    border: 10px solid black;
10   margin: 10px 20px 30px 40px;
11   padding: 10px 20px;
12 }
13 </style>
14 </head>
15 <body>
16
17 <div>this is div.</div>
18
19 </body>
20 </html>
21
22
23
```

Width and Height of an Element

- In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

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

- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

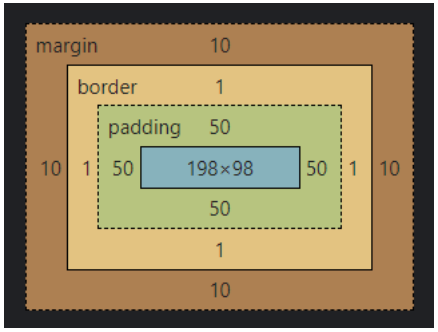
```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Total width :

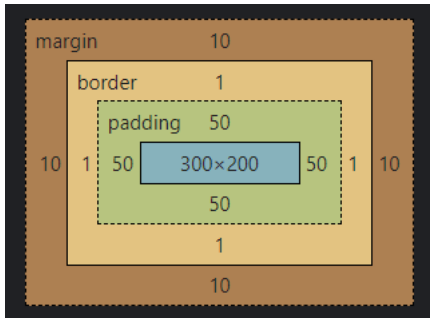
320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

Box-Sizing Property

- The CSS box-sizing property allows us to include the padding and border in an element's total width and height.
- box-sizing is 'border-box'



- box-sizing is 'content-box'




```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  .div2 {
6    width: 300px; height: 200px;
7    padding: 50px;
8    border: 1px solid red;
9    margin: 10px;
10   box-sizing: border-box;
11 }
12 </style>
13 </head>
14 <body>
15
16 <div class="div2">Hooray!</div>
17
18 </body>
19 </html>
20
21
22
23
```

CSS POSITION

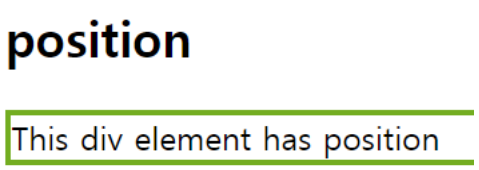
CSS Position

- **The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).**

- **There are five different position values:**
  - static (default)
  - relative
  - fixed
  - absolute

Position: Static

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- It is always positioned according to the normal flow of the page.



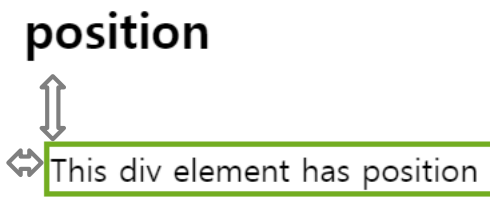

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  position: static;
  left: 10px; top: 20px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position</h2>
<div>
This div element has position
</div>

</body>
</html>
```

Position: Relative

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.



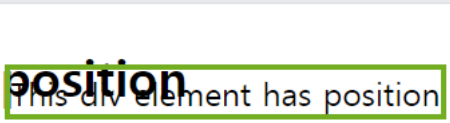

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  position: relative;
  left: 10px; top: 20px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position</h2>
<div>
This div element has position
</div>

</body>
</html>
```

Position: Fixed

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled



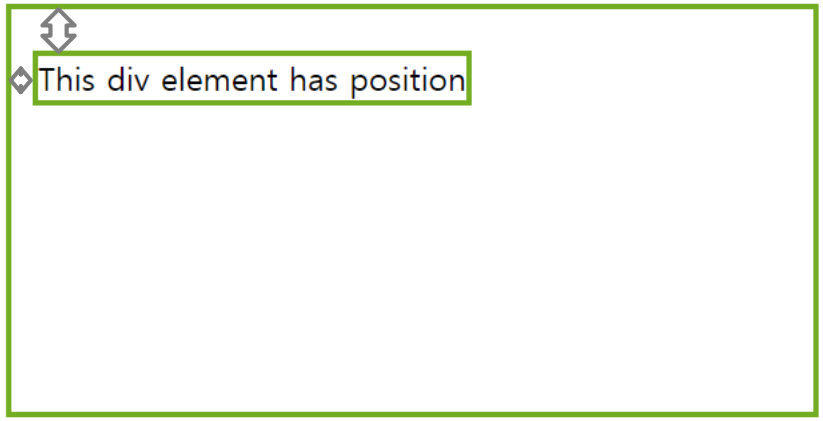

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  position: fixed;
  left: 10px; top: 20px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position</h2>
<div>
This div element has position
</div>

</body>
</html>
```

Position: Absolute

- An element with position: absolute; is positioned relative to the nearest positioned (relative / fixed/ absolute) ancestor.
- if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling

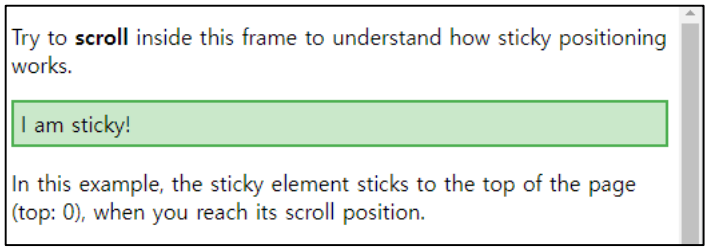


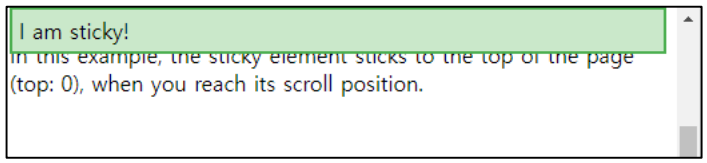

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px; height: 200px;
  border: 3px solid #73AD21;
}
div {
  position: absolute;
  left: 10px; top: 20px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
<div class="relative">
    <div>This div element has
position</div>
</div>
</body>
</html>
```

Position: Sticky

- An element with position: sticky; is positioned based on the user's scroll position.

- A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).






```
<head>
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
<body>
<p>Try to <b>scroll</b> inside
this frame to understand how
sticky positioning works.</p>
<div class="sticky">I am
sticky!</div>
<div style="padding-
bottom:2000px">
  <p>…</p>
</div>
…
```

CSS FLOAT AND CLEAR

The float Property

- **The float property is used for positioning and formatting content**
  - e.g. let an image float left to the text in a container

- **The float property can have one of the following values:**
  - left- The element floats to the left of its container
  - right- The element floats to the right of its container
  - none - The element does not float (will be displayed just where it occurs in the text). This is default
  - inherit - The element inherits the float value of its parent
- **In its simplest use, the float property can be used to wrap text around images.**

Float Example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
        float: right;
}
</style>
</head>
<body>
<h2>Float</h2>
<p><img src="pineapple.jpg" >
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Phasellus imperdiet, nulla et
dictum interdum, nisi lorem
egestas odio, vitae scelerisque
enim ligula venenatis dolor.</p>
</body>
</html>
```

Float Example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
        float: left;
}
</style>
</head>
<body>
<h2>Float</h2>
<p><img src="pineapple.jpg" >
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Phasellus imperdiet, nulla et
dictum interdum, nisi lorem
egestas odio, vitae scelerisque
enim ligula venenatis dolor.</p>
</body>
</html>
```

Float Example

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
        float: none;
}
</style>
</head>
<body>
<h2>Float</h2>
<p><img src="pineapple.jpg" >
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Phasellus imperdiet, nulla et
dictum interdum, nisi lorem
egestas odio, vitae scelerisque
enim ligula venenatis dolor.</p>
</body>
</html>
```

The clear Property

- **The clear property specifies what elements can float beside the cleared element and on which side.**

- **The clear property can have one of the following values:**
  - none - Allows floating elements on both sides. This is default
  - left - No floating elements allowed on the left side
  - right- No floating elements allowed on the right side
  - both - No floating elements allowed on either the left or the right side
  - inherit - The element inherits the clear value of its parent

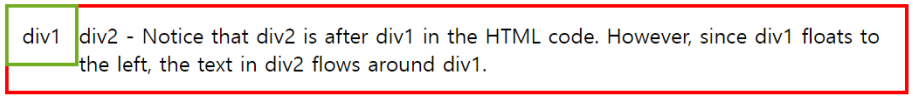
- **The most common way to use the clear property is after you have used a float property on an element.**

Clear Example

❖ **When clearing floats, you should match the clear to the float:**

- If an element is floated to the left, then you should clear to the left.
- Your floated element will continue to float, but the cleared element will appear below it on the web page.

Without clear




```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
  float: left;
  padding: 10px;
  border: 3px solid #73AD21;
}
.div2 {
  padding: 10px;
  border: 3px solid red;
}
</style>
</head>
<body>
<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice
that ...</div>
</body>
</html>

```

Clear Example

❖ **When clearing floats, you should match the clear to the float:**

- If an element is floated to the left, then you should clear to the left.
- Your floated element will continue to float, but the cleared element will appear below it on the web page.

With clear

div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  .div3 {
6    float: left;
7    padding: 10px;
8    border: 3px solid #73AD21;
9  }
10 .div4 {
11   padding: 10px;
12   border: 3px solid red;
13   clear: left;
14 }
15 </style>
16 </head>
17 <body>
18 <h2>With clear</h2>
19 <div class="div3">div3</div>
20 <div class="div4">div4 - Here,
21 clear: left; ...</div>
22 </body>
23 </html>
```

Z-INDEX PROPERTY

The z-index Property

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- z-index only works on positioned elements (position: absolute, position: relative, position: fixed, or position: sticky)


```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<img src="img_tree.png">
<p>Because the image has a z-
index of -1, it will be placed
behind the text.</p>
</body>
</html>
```

OVERFLOW

Overflow

❖ **The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.**

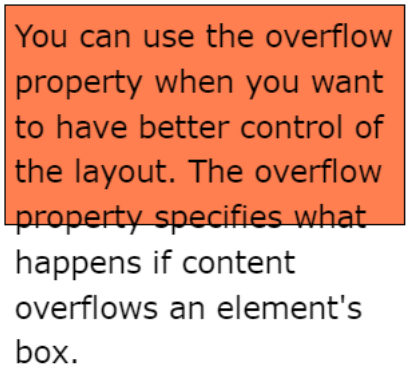
❖ **The overflow property has the following values:**

- visible - Default. The overflow is not clipped. The content renders outside the element's box
- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
- auto - Similar to scroll, but it adds scrollbars only when necessary

❖ **The overflow property only works for block elements with a specified height.**

Overflow: visible

❖ By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box.




```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: coral;
  width: 200px;
  height: 65px;
  border: 1px solid;
  overflow: visible;
}
</style>
</head>
<body>
<div>You can use the overflow
property when you want to have
better control of the layout. The
overflow property specifies what
happens if content overflows an
element's box.</div>

</body>
</html>
```

Overflow: hidden, scroll, auto

❖ hidden

- With the hidden value, the overflow is clipped, and the rest of the content is hidden:

❖ scroll

- Setting the value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box.
- Note that this will add a scrollbar both horizontally and vertically (even if you do not need it)

❖ auto

- The auto value is similar to scroll, but it adds scrollbars only when necessary: