INTRODUCTION TO WEB DEVELOPMENT

Code: COMP07009

Week 5

Reference: http://www.w3schools.com

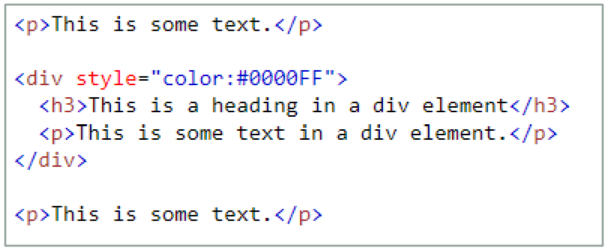
CSS CONTINUED

Layout

<div> Tag

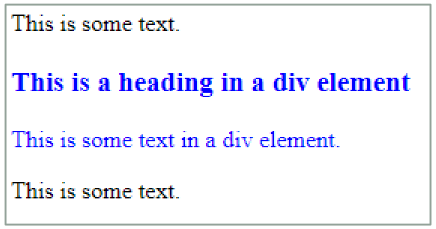
- Defines a division or a section in an HTML document
- It is used to group block elements to format them using CSS
- It is very often used together with CSS, to layout a web page

- Html code



```
<p>This is some text.</p>

<div style="color:#0000FF">
  <h3>This is a heading in a div element</h3>
  <p>This is some text in a div element.</p>
</div>

<p>This is some text.</p>
```

Output in browser



This is some text.

This is a heading in a div element

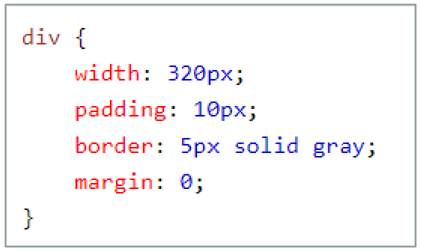
This is some text in a div element.

This is some text.

UNIVERSITY OF THE WEST of SCOTLAND UWS

Width and Height of <div>

- In order to style a <div> element to have a total width of 350px
- Example



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

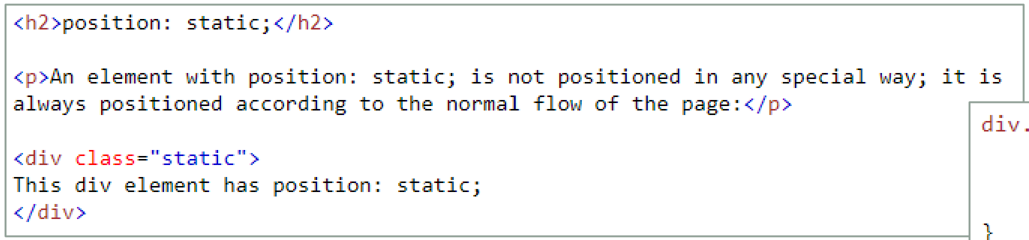
- Based on the Box model
- 320px (width)
- + 20px (left and right padding)
- + 10px (left and right border)
- + 0 px (left and right margin) **= 350 px**

CSS Positioning

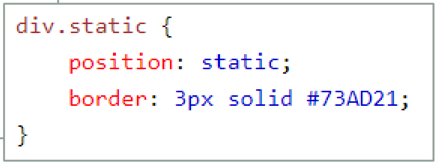
- The position property specifies the type of positioning method used for an element
- Static, relative, fixed or absolute
- Elements are then positioned using the top, bottom, left and right properties.
- These properties will not work unless the position property is first set.
- They also work differently depending on the position value.

Position – static

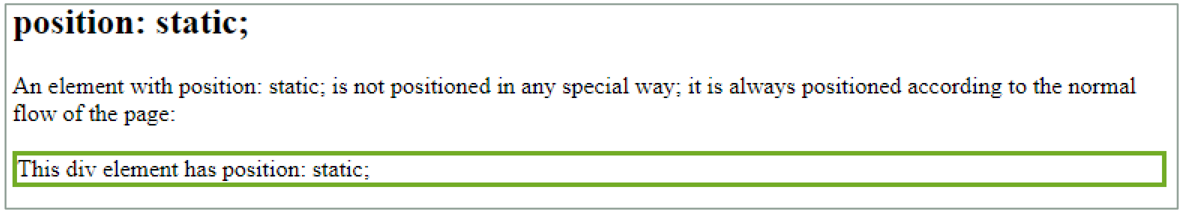
- HTML elements are positioned static by default
- Static positioned elements are not affected by top, bottom, left or right properties
- It is always positioned according to the normal flow of the page



```
<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
This div element has position: static;
</div>
```



```
div.static {
    position: static;
    border: 3px solid #73AD21;
}
```



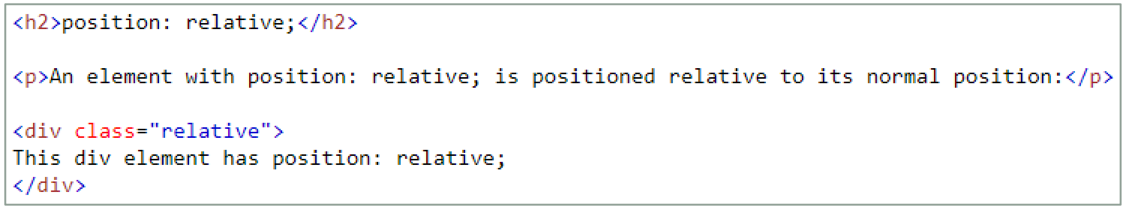
position: static;

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

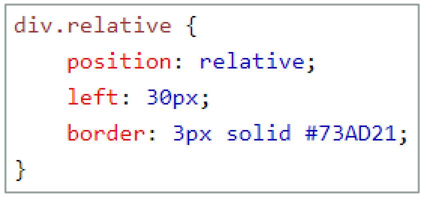
This div element has position: static;

Position – relative

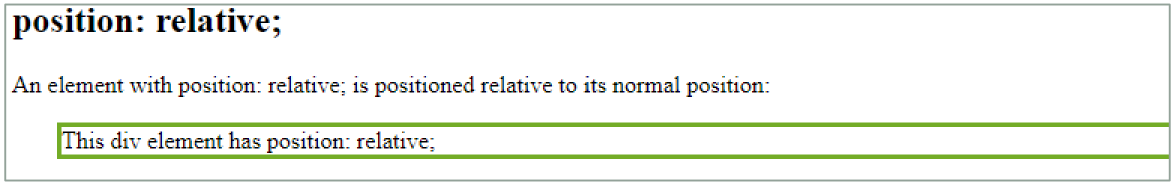
- HTML elements are positioned relative to their normal position
- Setting the top, bottom, left or right properties will cause it to be adjusted away from its normal position
- Other content will not be adjusted to fit into any gap left by the element page



```
<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to its normal position:</p>

<div class="relative">
This div element has position: relative;
</div>
```



```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
}
```



position: relative;

An element with position: relative; is positioned relative to its normal position:

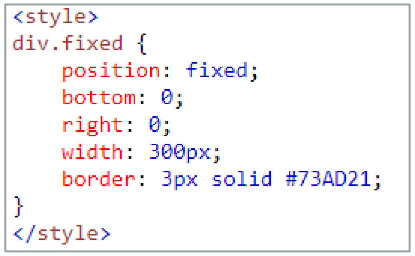
This div element has position: relative;

Position – fixed

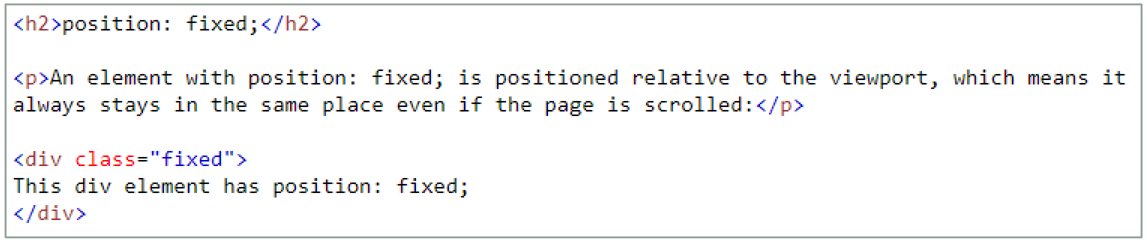
- HTML elements are positioned relative to the viewport
- This means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.
- The **viewport** is the user's visible area of a web page. The **viewport** varies with the device, and will be smaller on a mobile phone than on a computer screen.

Position – fixed

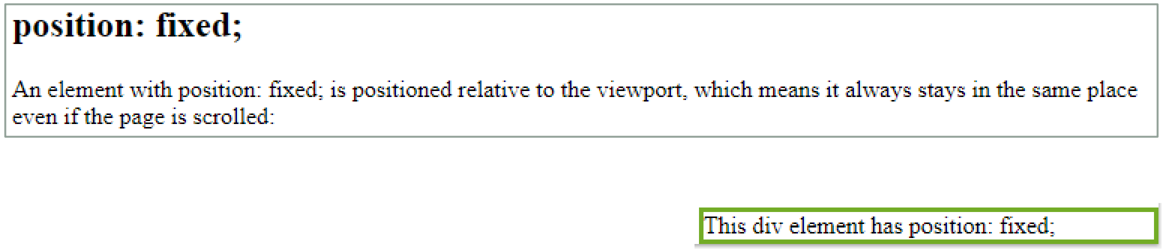
- The fixed element in the lower-right corner of the page has been set by using fixed positioning, regardless where the <h2> and <p> elements are displayed



```
<style>
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
</style>
```



```
<h2>position: fixed;</h2>

<p>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:</p>

<div class="fixed">
This div element has position: fixed;
</div>
```



position: fixed;

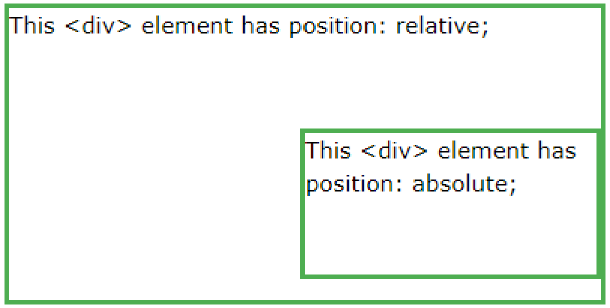
An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

This div element has position: fixed;

Position – absolute

- An element is positioned relative to the nearest positioned ancestor (ie. the closest ancestor to the element with a set position value, other than static - which is default).
- A "positioned" element is one whose position is anything except static.

```
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```


This <div> element has position: relative;

This <div> element has position: absolute;

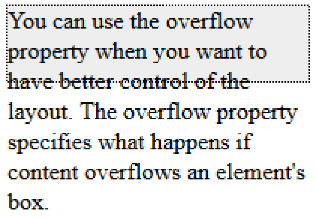
```
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
```

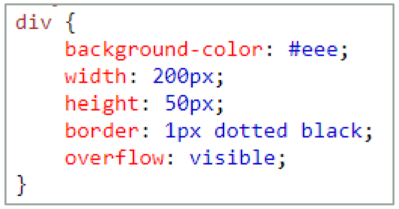
Overflow

- The overflow property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area.
- It has the different values
- Visible – Default
- Hidden – the overflow is clipped and rest of content is invisible
- Scroll – the overflow is clipped but a scrollbar is added to see the rest of the content
- Auto – if the overflow is clipped, a scrollbar is added to see the rest of the content

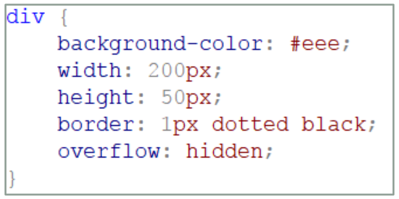
Overflow

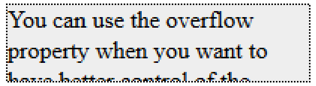
- Visible


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.

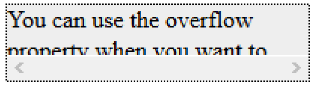

```
div {
    background-color: #eee;
    width: 200px;
    height: 50px;
    border: 1px dotted black;
    overflow: visible;
}
```

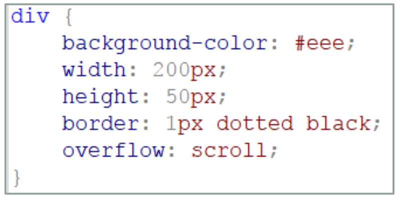
- Hidden


```
div {
    background-color: #eee;
    width: 200px;
    height: 50px;
    border: 1px dotted black;
    overflow: hidden;
}
```


You can use the overflow property when you want to

- Scroll


You can use the overflow

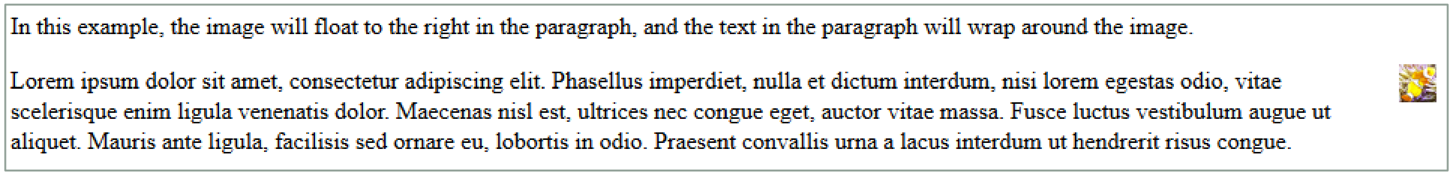

```
div {
    background-color: #eee;
    width: 200px;
    height: 50px;
    border: 1px dotted black;
    overflow: scroll;
}
```

Float

- The float property can be used to wrap text around images

```
img {
    float: right;
    margin: 0 0 5px 5px;
}
```

```
<p>In this example, the image will float to the right in the paragraph, and the
text in the paragraph will wrap around the image.</p>

<p><img src="images/fish.jpg" alt="fish" width="25px" height="25px">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim
ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor
vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula,
facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus
interdum ut hendrerit risus congue. </p>
```



In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

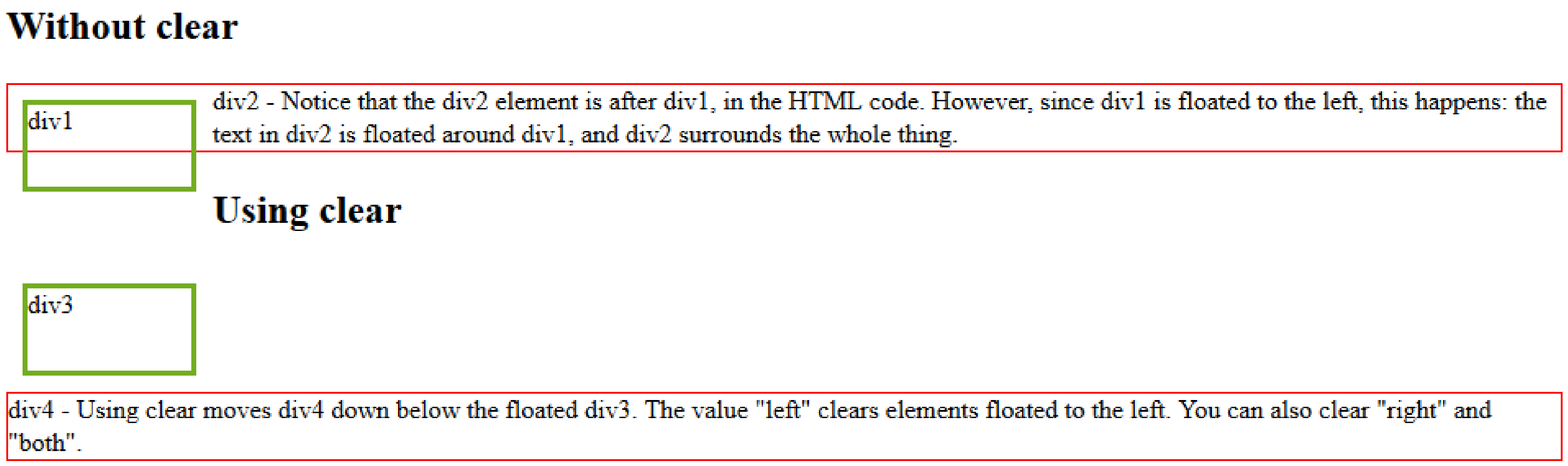
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue.

Clear

- The clear property is used to control the behavious of floating elements
- Elements after a floating element will flow around it.
- To avoid this, use the clear property.
- The clear property specifies on which side of an element floating elements are not allowed to float

Clear

- The clear property is used to control the behavious of floating elements
- Elements after a floating element will flow around it.
- To avoid this, use the clear property.



Without clear

div1

div2 - Notice that the div2 element is after div1, in the HTML code. However, since div1 is floated to the left, this happens: the text in div2 is floated around div1, and div2 surrounds the whole thing.

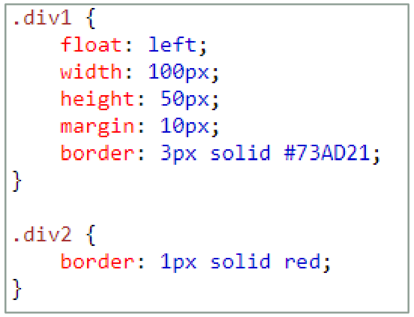
Using clear

div3

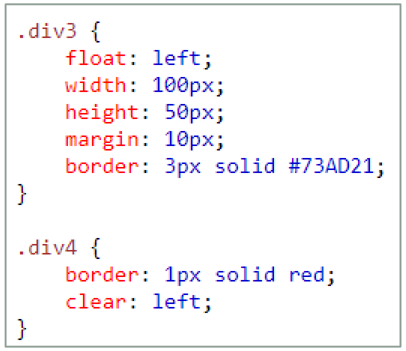
div4 - Using clear moves div4 down below the floated div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

Clear

- CSS code

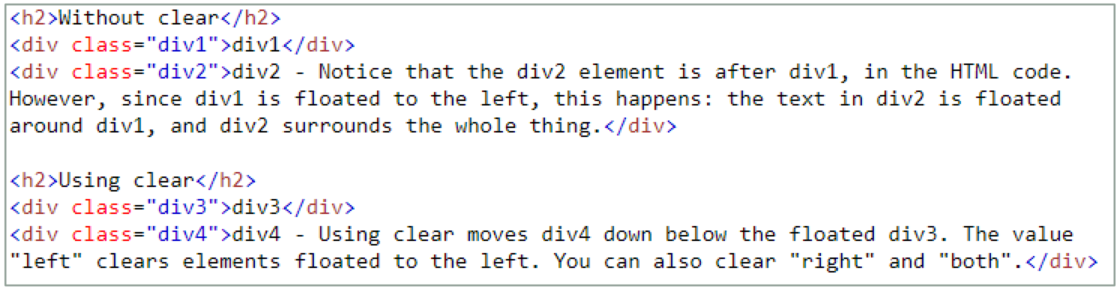


```
.div1 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}

.div2 {
    border: 1px solid red;
}
```



```
.div3 {
    float: left;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}

.div4 {
    border: 1px solid red;
    clear: left;
}
```

- HTML code



```
<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that the div2 element is after div1, in the HTML code.
However, since div1 is floated to the left, this happens: the text in div2 is floated
around div1, and div2 surrounds the whole thing.</div>

<h2>Using clear</h2>
<div class="div3">div3</div>
<div class="div4">div4 - Using clear moves div4 down below the floated div3. The value
"left" clears elements floated to the left. You can also clear "right" and "both".</div>
```