



WEBD 162

Class 10
Chapter 15 - Floating and Positioning

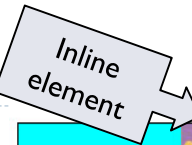
1

Normal Flow

<p id="p1">This is para 1 etc</p>

<p id="p2">This is para 2 etc</p>

<p id="p3">This is para 3 etc</p>



This is para 1 etc... This is para 1 etc...
This is para 1 etc... This is para 1 etc... This is para 1 etc...
This is para 1 etc... This is para 1 etc... This is para 1 etc...
This is para 1 etc... This is para 1 etc... This is para 1 etc...

This is para 2 etc This is para 2 etcThis is para 2 etc
This is para 2 etcThis is para 2 etcThis is para 2 etc

This is para 3 etc! This is para 3 etc! This is para 3 etc!
This is para 3 etc! This is para 3 etc! This is para 3 etc!
This is para 3 etc! This is para 3 etc! This is para 3 etc!
This is para 3 etc!

Block elements

2

Floating – Place elements Side by Side

▶ float property

- ▶ left
 - ▶ Places element on left most side
- ▶ right
 - ▶ Places element on right most side
- ▶ none
 - ▶ Removes the float

▶ clear property – to stop the floating action

- ▶ left
- ▶ right
- ▶ both

3

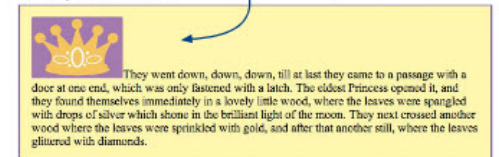
Floating an **inline** element – pg 343

<p> They
went down, down, down ...

</p>

Inline image in the normal flow.

Space next to the image is held clear.



img { float: right; }

Inline image floated to the right.

image could use
some space

float breaks
naturally

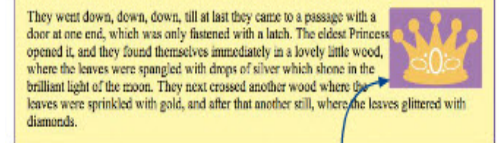


Image moves over and text wraps around it.

4

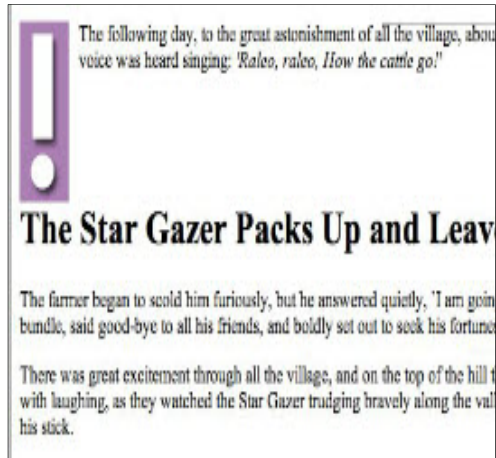
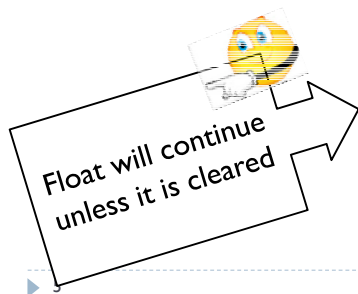
Clearing a float – pg 347

<p> the following day...</p>

<h2>The Star Gazer ..</h2>

```
img { float: left; }
```

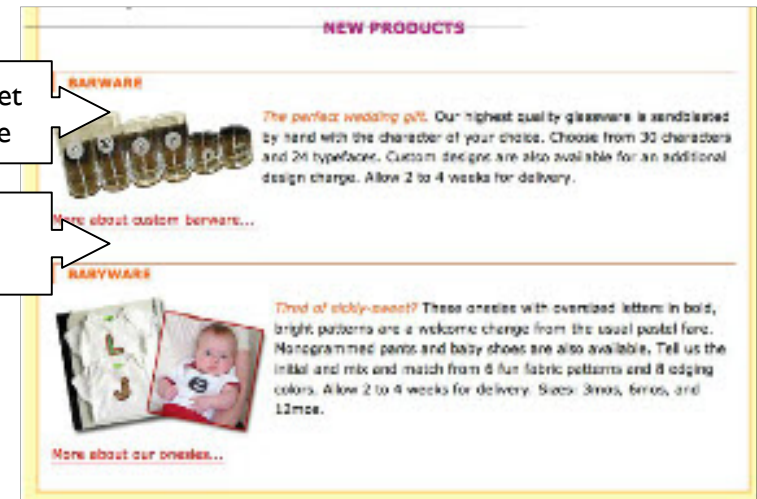
```
h2 { clear: left; }
```



Clearing a Float – pg 348

Float is set on image

Float is cleared



▶ 6

Floating a **block** element

- ▶ Need a **width** on at least one element
- ▶ Vertical margins are collapsed – you can't add **top/bottom margin**
- ▶ May need to **clear** / stop the float, depending on the layout

Floating a **block** element – pg 346

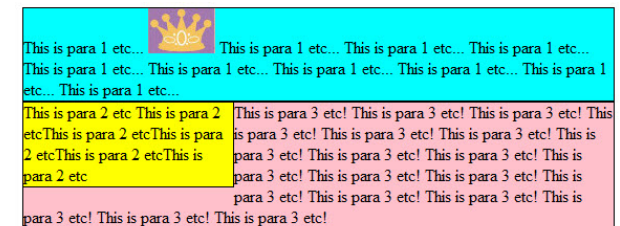
<div id="d1">This is para 1 etc</div>

<div id="d2">This is para 2 etc</div>

<div id="d3">This is para 3 etc</div>

```
#d2 { width: 200px; float: left; }
```

unwanted effect



▶ 8

▶ 7

Floating a **block** element

How to fix the previous slide

- ▶ Set a **left margin** on the right element that is at least as **wide** as the left container

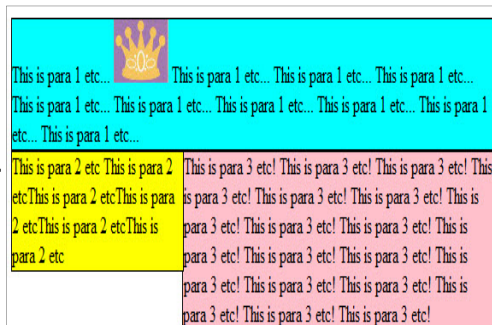
```
<div id="d1">This is para 1 etc</div>
```

```
<div id="d2">This is para 2 etc</div>
```

```
<div id="d3">This is para 3 etc</div>
```

```
#d2 { width: 200px; float: left; }
```

```
#d3 { margin-left: 200px; }
```



▶ 9

Floating a **block** element

How to fix the previous slide

- ▶ set **width** on both containers
- ▶ **float** both container to **left**

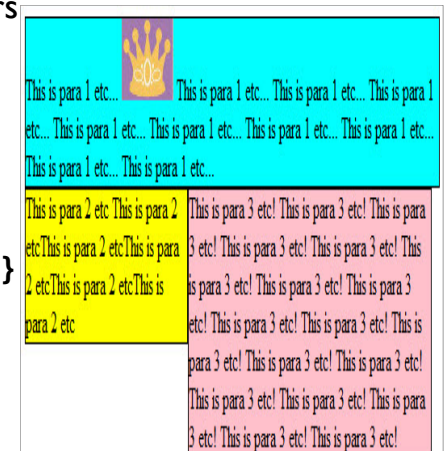
```
<div id="d1">This is para 1 etc</div>
```

```
<div id="d2">This is para 2 etc</div>
```

```
<div id="d3">This is para 3 etc</div>
```

```
#d2 { width: 200px; float: left; }
```

```
#d3 { width: 300px; float: left; }
```



▶ 10

Floating a **block** element – **right and left**

How to fix the previous slide

- ▶ set **width** on both containers
- ▶ set **float** both container

```
<div id="d1">This is para 1 etc</div>
```

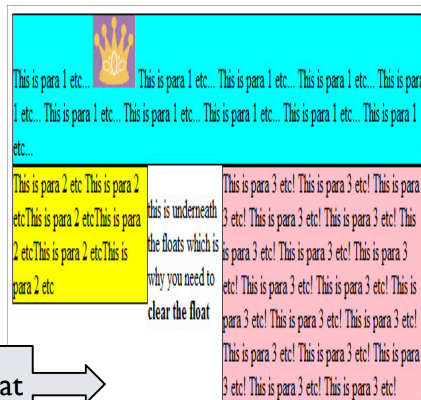
```
<div id="d2">This is para 2 etc</div>
```

```
<div id="d3">This is para 3 etc</div>
```

```
<p>This is underneath...</p>
```

```
#d2 { width: 200px; float: left; }
```

```
#d3 { width: 300px; float: right; }
```



Need to clear the float

▶ 11

Clearing a float – **clear fix**

How to clear the float on previous slide

```
<div id="d1">This is para 1 etc</div>
```

```
<div id="d2">This is para 2 etc</div>
```

```
<div id="d3">This is para 3 etc</div>
```

```
<p class="clearfix">This is underneath...</p>
```

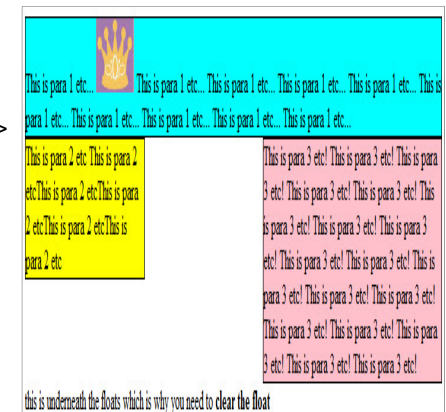
```
.clearfix:after {
```

```
content: "";
```

```
display: block;
```

```
clear: both;
```

```
}
```



▶ 12

Floating **multiple** elements – **navigation** pg 351

```
<ul>
<li><a href="#">serif</a></li>
<li><a href="#">sans-serif</a></li>
<li><a href="#">script</a></li>
<li><a href="#">display</a></li>
<li><a href="#">dingbats</a></li>
</ul>
```

```
ul li a {
  display: block;
  width: 100px;
  text-align: center;
  text-decoration: none;
  padding: 10px 10px;
  margin: 5px;
  color: white;
  background-color: lightslategray;
  border-top-left-radius: 15px;
  border-top-right-radius: 15px;
}

ul li {
  float: left;
}
```

serif

sans-serif

script

display

dingbats

▶ 13

Floating - containing floats – pg 352



Hand-stitched iPhone motif
[More info](#)



```
#container {
  overflow: auto;
  width: 100%;
  background-color: #GGG;
  padding: 1em;
}
```



Hand-stitched iPhone motif
[More info](#)



▶ 14

Positioning – pg 356

- ▶ **Static**
 - ▶ Default
 - ▶ No positioning
- ▶ **Relative**
 - ▶ Relative to it's original position
- ▶ **Absolute**
 - ▶ Removed from the default flow
 - ▶ Positioned to the browser window or parent container (parent must have positioning to do this)
- ▶ **Fixed**
 - ▶ Stays in one place when the page scrolls

▶ 15

Positioning – pg 356

- ▶ **top: value;**
- ▶ **bottom: value;**
- ▶ **right: value;**
- ▶ **left: value;**

▶ 16

Containing blocks – pg 360

► Containing block element

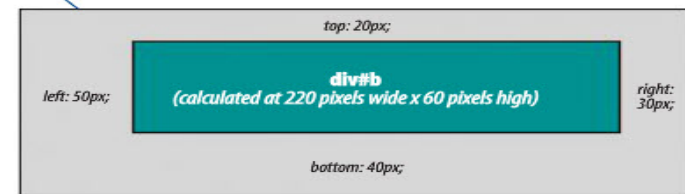
- If the element is positioned within an containing (parent) element has positioning, then that element is positioned to the container block
- If not, the element is positioned to it's parent

► 17

Common use of Positioning – pg 363

```
div#a {  
  position: relative; /* creates the containing block */  
  height: 120px;  
  width: 300px;  
  border: 1px solid;  
  background-color: #CCC;  
}  
  
div#b {  
  position: absolute;  
  top: 20px;  
  right: 30px;  
  bottom: 40px;  
  left: 50px;  
  border: 1px solid;  
  background-color: teal;  
}
```

div#a (width: 300px; height: 120px;)



► 18

Common use of Positioning

– drop down navigation



► 19

Stacking order – z-index– pg 368

z-index: numeric-value;

```
#C {  
  z-index: 1;  
  position: absolute;  
  top: 250px;  
  left: 225px;  
}
```

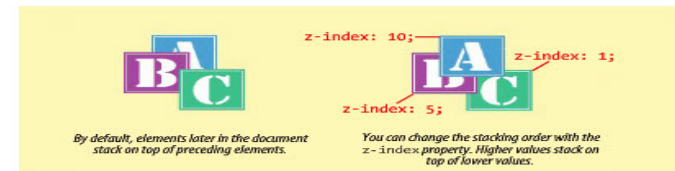


Figure 15-25. Changing the stacking order with the z-index property.

To be honest, the z-index property is not often required for most page layouts, but you should know it's there if you need it. If you want to guarantee that a positioned element always ends up on top, assign it a very high z-index value, such as:

```
img#essential {  
  z-index: 100;  
  position: absolute;  
  top: 0px;  
  left: 0px;  
}
```

► 20