ADV3550
Instructor: Genevieve Hitchings

Positioning CSS
The CSS positioning properties allow you to position elements. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.
There are four different positioning methods.

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

Fixed Positioning
An element with fixed position is positioned relative to the browser window.
It will not move even if the window is scrolled:
p.pos_fixed
{
position:fixed;
top:30px;
right:5px;
}

Relative Positioning
A relative positioned element is positioned relative to its normal position.
h2.pos_left
{
position:relative;
left:-20px;
}

h2.pos_right
{
position:relative;
left:20px;
}

The content of a relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

h2.pos_top
{
position:relative;
top:-50px;
}
Relatively positioned elements are often used as container blocks for absolutely positioned elements.

Absolute Positioning

An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

```
h2
{
position:absolute;
left:100px;
top:150px;
}
```

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.
Absolutely positioned elements can overlap other elements.

Overlapping Elements

When elements are positioned outside the normal flow, 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). An element with greater stack order is always in front of an element with a lower stack order.
An element can have a positive or negative stack order:

```
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1
}
```

What is CSS Float?

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it. Float is very often used for images, but it is also useful when working with layouts.

How Elements Float

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down. 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.
The elements after the floating element will flow around it.
The elements before the floating element will not be affected.
If an image is floated to the right, a following text flows around it, to the left:

```
img
{
float:right;
}
```

Floating Elements Next to Each Other

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

Here we have made an image gallery using the float property:

```
.thumbnail
{
float:left;
width:110px;
height:90px;
margin:5px;
}
```

Turning off Float - Using Clear
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.

Add a text line into the image gallery, using the clear property:
```
.text_line
{
clear:both;
}
```

CSS Horizontal Align
In CSS, several properties are used to align elements horizontally.

Aligning Block Elements
A block element is an element that takes up the full width available, and has a line break before and after it.
Examples of block elements:
```
<h1>
<p>
<div>
```

Center Aligning Using the margin Property
Block elements can be aligned by setting the left and right margins to "auto". Setting the left and right margins to auto specifies that they should split the available margin equally. The result is a centered element:
```
.center
{
margin-left:auto;
margin-right:auto;
width:70%;
background-color:#b0e0e6;
}
```
(Tip: Aligning has no effect if the width is 100%).

Left and Right Aligning Using the position Property
One method of aligning elements is to use absolute positioning:
```
.right
{
position:absolute;
right:0px;
width:300px;
background-color:#b0e0e6;
}
```