



Introduction

Welcome to lesson eight! This lesson is all about **CSS floats**. The CSS `float` property is very powerful. It allows us to push elements to the left, or right, produce elegant menus, and even control entire page layouts. However, floating can be really frustrating, if you don't understand how it works. So let's take a detailed look at the nature of floats.

Nature of Floats

A floated element is positioned within the normal flow of a document, and then taken out of the flow. It is then shifted to the left, or the right on the current line, as far as possible. A floated element will shift to the left, or right and rest against one of three different things:

- The first thing it will come to rest against is the edge of the viewport.
- Secondly, the element could come to rest against the edge of the parent container.
- The third option, is that it could come to rest against the edge of another floated item.

The `float` property allows the following values:

- `left` - will take the element out of flow, and shift it to the left on the current line.
- `right` - will take the element out of flow, and shift it to the right on the current line.
- `none` - will leave the element in normal flow.
- `inherit` - will inherit the float behavior from the parent element.

Exercise

We're going to focus on two images inside the `div.container`. They all sit in normal flow, so the paragraphs aren't interacting with these elements.

```
.example01 { float: left }
```

The image is now set to float left, and the paragraph then slides up underneath the image. However, the line boxes inside the paragraph shortened, so the text doesn't overlap, or go under the image. One way you can test to see whether paragraph is really going under the image, is to give it a border:

```
p { border: 5px solid red }
```

now that's going to put a border around every paragraph on the page. Now if we go, and reload, you can see that this is clearly sitting out underneath that image. Another thing we want to test, is what will happen if we put margin around this image.

```
.example01  
{  
    float: left;  
    margin-right: 20px;  
}
```

You will see that a margin is applied to that image, and it's pushed away the line boxes. So it hasn't affected the paragraph itself - it's affected each of line boxes, that are interacting with that image.

```
.example02  
{  
    float: right;  
    margin-left: 20px;  
}
```

The second image will move over to the right edge of the container, and the paragraph like the other one, has now slid up underneath the image. The line box is also shortened to accommodate the margin associated with that image.