



Absolute Positioning

Now let's look at **absolute positioning**. This can be written with

```
div {position: absolute}
```

When given absolute positioning, an element will move out of flow up off the page. In-flow content that comes after absolutely positioned element will ignore it and re-flow accordingly.

An absolutely positioned element will be positioned relative to the nearest ancestor with positioning. However, if no ancestors have positioning, the element will be positioned relative to the viewport.

By default, elements with absolute positioning have auto height and widths, so it will shrink wrap around the content. However width and height can be redefined.

Absolutely positioned elements can also be stretched. If an absolutey-positioned element is set to `left: 0` and `right: 0`, it will stretch to fit the width of the parent element or the viewport, if there's no positioning on the parent element.

If an absolutely positioned element is set to `top: 0` and `bottom: 0`, it will stretch to the height of the parent element or the viewport.

Exercise

In this exercise we will focus on h2 element. First of all, assign positioning to it:

```
h2 { position: absolute; }
```

h2 will go up off the page. The content below it will ignore this element now. h2 will also be shrink wrapped.

```
h2
{
  left: 0;
  top: 0;
}
```

h2 will jump to the top of the page - it's actually positioning itself in the very top left corner of the viewport. What's happening is, it's looking for a parent element with positioning. If it can't find any, it will then position itself relative to the viewport.

h2 has a `div.container` as a parent and we can assign it with some positioning.

```
.container { position: relative; }
```

You will see that the h2 is now positioned relative to the container rather than to the viewport.

```
h2 { left: 20px; }
```

The element will be shifted 20px to the right.

```
h2 { top: 20px; }
```

In this case h2 will be shifted downwards 20px.

Now we're going to try and stretch it:

```
h2 {  
  left: 0;  
  right: 0;  
}
```

The element will be stretched to the full width of the parent container.