The z-index property
================

1. The z-index property decides which element should be given priority if overlapping occurs.

2. By default z-index is 0 for every element so browsers uses STACK principle to resolve overlapping i.e. the element occuring after will overlap the element occuring before. In other words the 1st element will be overlapped by the 2nd, the 2nd element will be overlapped by the 3rd and so on.

3. To change this behaviour we can set z-index of an element to a greater value.

CSS CODE
=========

```
#one {
        width: 300px;
        border: 1px solid #000;
        padding: 20px;
        background: lightgreen;
        position: relative;
        top: 50px;

      }

      #two {
        width: 300px;
        border: 1px solid #000;
        padding: 20px;
        background: lightpink;
        position: relative;
        left: 30px;
      }
```

HTML CODE
==========

```
<div id="one">
      This is some text inside the first div element and is lengthy also.
    </div>
    <div id="two">
      This is some text inside the second div element and is lengthy also.
    </div>
```

This div will overlap the first div

```
CSS CODE
=========
#one {
    width: 300px;
    border: 1px solid #000;
    padding: 20px;
    background: lightgreen;
    position: relative;
    z-index: 1;
    top: 50px;

   }

   #two {
    width: 300px;
    border: 1px solid #000;
    padding: 20px;
    background: lightpink;
    position: relative;
    left: 30px;
```

```
HTML CODE
==========
<div id="one">
   This is some text inside the first div element and is lengthy also.
  </div>
  <div id="two">
   This is some text inside the second div element and is lengthy also.
  </div>
```

This div will overlap the second div because of gr z-index value of

Absolute Positioning

===============

1. Absolute positioning means as a programmer we specify exact coordinates for the element.

2. An absolutely position element is taken out of the normal flow after its position has been set.

CSS CODE
=========

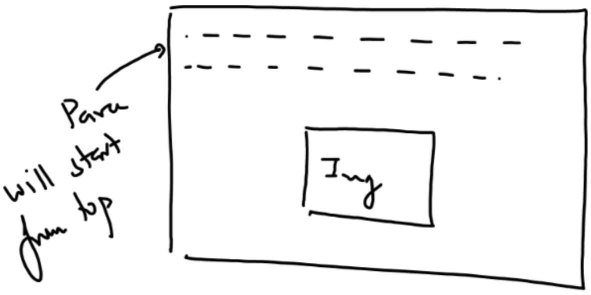
```
p {
    background-color: skyblue;

   }

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

HTML CODE
==========

```
<img src="/media/images/bgimg.jpg" />
  <p>
   Lorem ipsum dolor sit amet consectetur
adipisicing elit. Sunt nisi odio
   pariatur......
 </p>
```



Fixed positioning is same as absolute but the only difference is that a fixed positioned elelement does not move from it's location when the page is scrolled.

Example
========

```
h1 {
     color: red;
     position: fixed;
     left: 640px;
     top: 500px;
    }
```

Sticky positioning
==============

A sticky positioned elemenet will initially behave like a relative element but if we keep on scrolling the page the browser will make them fixed as soon as their values are matched. In short we can say sticky is relative + fixed.

Example
======

```
table {
  text-align: center;
  position: sticky;
  top: 0px;
}
```