CSS Positioning

In the previous chapter you created a web page layout using positioning with floats. This method is the most popular method used today in web design.
There is another method of placing elements on the web page – using CSS position. CSS positioning allows you to precisely position an element using exact coordinates.

In this chapter we will briefly discuss three types of CSS positioning: absolute, relative and fixed:

Absolute Positioning

Absolute positioning creates an element that behaves as a "a free agent", independently from the rest of the document.

If an element is nested inside of another element, its position will be relative to the top left-hand corner of the parent element.

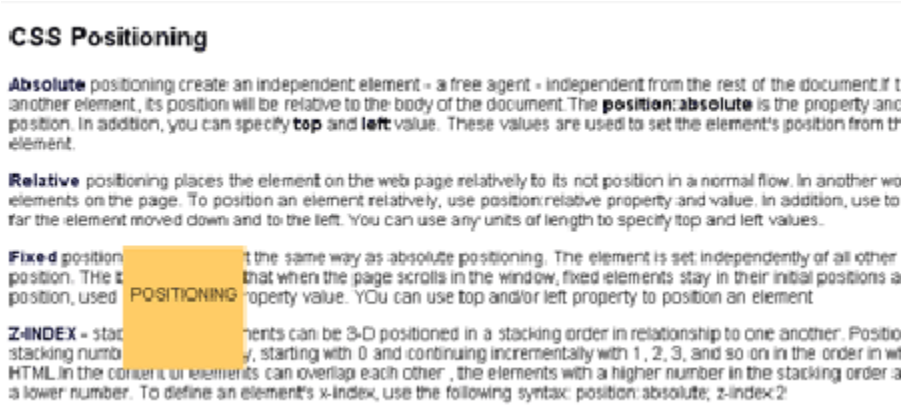
If an element is not nested inside of another element, its position will be relative to the top left-hand corner of the browser window.

To set the absolute position, use **position:absolute** property/value. In addition, you can specify **top** and **left** value. These values are used to set the element's position from the top and left edges of its parent element. You can use any units of length to specify values for the **top** and **left** properties. You can specify negative values for **top**and **left** properties.

Example:

```
#yellowBox {
position:absolute;
 left:100px;
 top:200px
}
```

Below you can see element (**#yellowBox**) positioned absolutely. This element is moved 100px to the right and 200px down from the top left corner of the window (parent element)



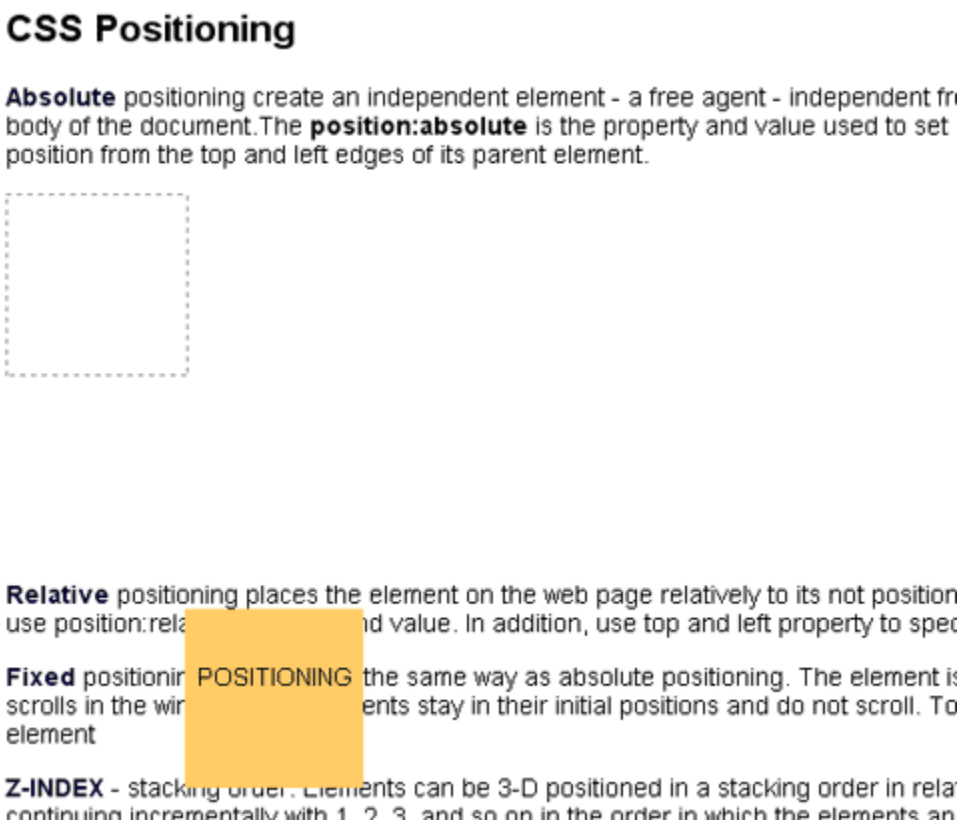
Relative Positioning

Relative positioning places the element on the web page relatively to its natural position in a normal flow. In another words, relative position offsets elements on the page. To position an element relatively, use **position:relative** property and value. In addition, use **top** and l**eft** property to specify how far the element should move down and to the left. You can use any units of length to specify **top** and **left** values. You can use negative values for top and left properties.

Example:

```
#yellowBox {
position:relative;
left:100px;
top:200px;
}
```

In the picture below, you can see the same element positioned relatively. Notice it was offset from its normal position (shown as a dashed box). The space that the element occupied in normal flow is empty.



Fixed Positioning

Fixed positioning does almost the same as absolute positioning. The element is set independently of all other content on the page in a specific position. The big difference is that when the page scrolls in the window, fixed elements stay in their initial positions and do not scroll with the page content. To set up a fixed position, used **position:fixed;** property and value. You can use **top** and/or **left** property to position an element precisely on the page.

Stacking Order (z-index)

CSS **z-index** property determines an element's stacking order. Elements can be 3-D positioned in a stacking order in relationship to one another.

Positioned elements are assigned stacking numbers automatically, starting with 0 and continuing incrementally with 1, 2, 3, and so on in the order in which the elements appear in the HTML.

If elements overlap each other, the element with a higher number in the stacking order appears over the elements that have a lower number.

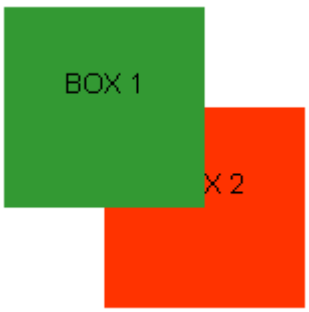
To define an element's z-index, use the following syntax:
position:absolute; z-index:*integer*

Example:

```
#box_green {
background-color:#339933;
position:absolute;
top: 500px;
left: 100px;
z-index:100;
}

#box_red {
background-color:#FF3300;
position:absolute;
top:550px;
left:150px;
z-index:0;
}
```

The Figure below demonstrates the results. Green box (#box_green element) has a higher z-index value (100) and overlaps the red box (#box_red).



CSS Positioning is a key element of the Dynamic HTML (DHTML), where JavaScript and the HTML Document Object Model can