

CSS Positioning & Layout

Creating layouts and controlling elements

About CSS Layout

There are two methods for arranging items on the page using CSS, Floating and Positioning.

- **Floating** an element moves it to the left or right with text wrapping around the element.
- **Positioning** specifies the location of any element anywhere on the page with pixel precision.

Float

The float property moves an element to the left or right allowing any proceeding content to wrap around the element. Floats can be used to create multicolumn layouts, navigation and table like alignment.

The values for float are left, right, none and inherit, with none being the default.

- * Float None
- * Float Left
- * Float Right
- * Floating Text
- * Clearing Floating Elements

Positioning

There are two main types of positioning: absolute and relative and a third that is slightly different called fixed. The basic CSS structure for all is:

```
tag {position: choice; top: 0px; bottom: 0px; left: 0px;  
right: 0px; }
```

Relative

All elements have a natural flow (very Zen) basically where you place it in the code, that is where it generally goes. Most elements will position themselves naturally top to bottom and left to right. When we position something relatively, the surrounding elements are not affected; rather we position the element relative to where it would have naturally appeared.

```
img {position: relative; top: 50px; left: 100px; }
```

The CSS could be applied to the item, but when creating CSS for layout, it is a good idea to apply to a class or ID. So actually, I would write my CSS like this:

```
. photo {position: relative; top: 50px; left: 100px; }
```

Absolute

When you position something absolutely it will appear at the exact pixel you specify. Let's say I want to position an image at 50 pixels from the top and 100 pixels from the left. I would write the CSS like this:

```
img {position: absolute; top: 50px; left: 100px; }
```

The CSS could be applied to the item, but when creating CSS for layout, it is a good idea to apply to a class or ID. So actually, I would write my CSS like this:

```
. photo {position: absolute; top: 50px; left: 100px; }
```

Elements positioned with Absolute can overlap other elements (not always a bad thing). Also, if you want the layout to be flexible, you might want to use ems and percentages when you position.

Fixed

When you position something fixed it will stay stationary regardless of what the user does with the browser. So even if a user scrolls, the image will stay still. Fixed may not work with older browsers. I would write the CSS like this:

```
img {position: fixed; top: 50px; left: 100px; }
```

The CSS could be applied to the item, but when creating CSS for layout, it is a good idea to apply to a class or ID. So actually, I would write my CSS like this:

```
. photo {position: fixed; top: 50px; left: 100px; }
```

Resources

- Float Tutorial

<http://css.maxdesign.com.au/floatutorial/index.htm>

- W3C Schools Positioning

http://www.w3schools.com/css/css_positioning.asp

Leverage

- Nice and Free CSS Layout Templates

<http://www.mycelly.com/>

- Layout Gala

<http://blog.html.it/layoutgala/>

- Little Boxes

http://www.thenoodleincident.com/tutorials/box_lesson/boxes.html