**Lesson 6: Positioning Objects**

**The Set Up:**

***Preparation:***

Open Dreamweaver. Navigate to the “02formatting\_objects” folder you copied for Lesson 5. In Dreamweaver, click File > Open, then open the “positioning.html” document from within the “02formatting\_objects” folder. Read through the information below, then follow along with the “positioning” section of the “formatting\_objects.pdf” you used for Lesson 5. Make changes to the “positioning.html” document in Dreamweaver as you read. Once you’re done making changes, play around with the CSS declarations to gain a better understanding of how positioning works!

**The Learning:**

***Object Positioning***

Specifies the placement of elements on the page (i.e. allows you to place elements almost anywhere you want – you can even layer one element on top of another); this is how modern, [multi-column page layouts](http://www.w3schools.com/css/tryit.asp?filename=trycss_float6) are created; positioning CSS styles can also be used to wrap text around an image when <img> tags are contained within <p> tags.

* Object positioning is used in combination with **Margins, Padding and Borders** to create clean, organized, and precise page layouts

***Types of Positioning***

There are multiple TYPES of positioning that can be used when organizing a website’s layout. The “position” property determines how a website’s elements will be positioned within the page. This property must be set before any values will actually work. There are 4 major types of positioning:

1. Static – Default positioning; is positioned according to normal page flow
2. Relative – Cause an object to be repositioned in relation to its own normal position
3. Fixed – Cause an object to remain in its set position, even when the page is scrolled
4. Absolute – Cause an object to be repositioned in relation to its nearest ancestor

There are (usually) four different values available for positioning:

1. Left
2. Right
3. Top
4. Bottom

***The “Float” and “Clear” Properties***

There are many different properties that allow you to change the positioning of your elements, but [**“float”** and **“clear”**](https://www.w3schools.com/css/css_float.asp) are the most popular.

* The **“float”** property is often combined with the width, height, border, margin, and padding properties to produce multi-column layouts
* Adding the **“clear”** property to the next element in your CSS will specify which side of the previous element a new element is not allowed to float on. Using it will prevent your layout from getting jumbled when elements try to overlap one another

***Centering with “{Margin:Auto;}”***

Take a look under the property column in the chart below. Notice how the float and clear objects are able to position elements to the left, and to the right… but there is no option for centering. This is where the {margin:auto;} CSS declaration comes in handy!

* The **“margin:auto” declaration** will cause the element to be horizontally (left and right) centered within the page. The remaining space will be AUTOmatically be split equally between the left and right margin.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OBJECT** | **DEFINITION** | **PROPERTY** | **EXAMPLE** | **UNIT** | **DEFAULT** |
| Position | determines how a website’s elements will be positioned within the page | position position:staticposition:relativeposition:fixedposition:absolute | {position: fixed;} | N/A | static |
| [Float](https://www.w3schools.com/cssref/pr_class_float.asp) | specifies whether or not an element should float (i.e. be repositioned within the page)\*Absolutely positioned (we will learn more about this later) elements ignore the float property | floatfloat:leftfloat:right | {float:right;} | N/A | no float; elements appear consecutively from the top to the bottom of the page |
| [Clear](http://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clear) | Specifies on which side of an element floating elements are not allowed to float; controls the behaviour of floating elements\*Elements after a floating element will flow around it. To avoid this, use the clear property | clearclear:left\*clear:right\*\*left – stops the next element from floating to the left (and overlapping the element before)\*right – stops the next element from floating to the right (and overlapping the element before) | {clear:left}if your previous element was set to {float:left;} you would also {clear:left;}if your previous element was set to {float:right;} you would also {clear:right;} | N/A | no clear; if no float is specified, no clear is necessary |
| [Margin](https://www.w3schools.com/css/css_margin.asp) | as we already know, the margin property controls how much space exists between elements (on the top, bottom, left and right); the margin property can also be used to center elements | marginmargin:auto\*automatically applies as much margin to the left and right sides of the element, as necessary, in order to center it | {margin: auto} | N/A | zero spacing between elements |

**The Doing:**

***Code Academy Activity –*** [***CSS Positioning***](https://www.codecademy.com/en/courses/web-beginner-en-6merh/0/1?curriculum_id=50579fb998b470000202dc8b)

Activity Instructions: Visit the Code Academy website by clicking the hyperlink above. Complete lessons 1-5 (The Box Model), 6-12 (Margins, Borders, Padding), 13-16 (Clearing Elements), 17-20 (Absolute, Relative, and Fixed Positioning), and 21-25 (Review)

***Comprehension Question:***

* What is “object positioning”? Why is it important?
* What are the four different types of positioning? What is the difference between them?
* What are the two most popular properties to use when creating page layouts?
* What declaration can be used to center elements?