Day 6.
Understanding CSS Position

The position property is used to align the different elements in the HTML page. Position Property plays an important role to make high-quality web pages.

There are 5 position properties in CSS:

1. static (default)
2. relative
3. absolute
4. Fixed

1. Position: Static It is the default position value for the element. Under static position, elements are positioned according to the normal flow of the page.

Note: left, right, top, and bottom properties will not affect if the position is static.

2. Position: Relative In this case, the element remains in the normal flow of the document but left, right, top, and bottom affect it. Elements get shifted from their original position in the document creating vacant space and other elements may adjust themselves according to the vacant space left by the element.

3. Position: absolute - Absolute elements do not follow the normal flow document instead they position themselves relative to the closest positioned ancestor. Its final position is determined using the top, bottom, left, and right.

Note: The positioned element means an element having position property other than static.

These element does not occupy any space and other elements treat absolute elements like there is no element. Parent element should be positioned and position property other than absolute and If parent element is not positioned then absolute element position themselves according to nearest positioned ancestor. We generally set z-index when using absolute position.

4 Position: Fixed - Fixed element does not follow normal document flow and position themselves relative to <HTML> tag. This element always sticks to the screen.

CSS Transition

Transitions in CSS allow us to control the way in which transition takes place between the two states of the element. For example, when hovering your mouse over a button, you can change the background color of the element with help of a CSS selector and pseudo-class

There are four CSS properties that you should use, all or in part (at least two, transition-property and transition-duration, is a must), to animate the transition.

1. **Transition-property:** This property allows you to select the CSS properties which you want to animate during the transition(change).
2. **Transition duration:** This property allows you to determine how long it will take to complete the transition from one CSS property to the other.
3. **Transition-timing function:** This property allows you to determine the speed of change and the manner of change, during the transition. Like, the change should be fast at the beginning and slow at the end, etc.
4. **Transition-delay:** This property allows you to determine the amount of time to wait before the transition actually starts to take place.

The Shorthand Property You can combine all the four transition properties mentioned above, into one single shorthand property, according to the syntax given below. This saves us from writing long codes and prevents us from getting messy. Note the ordering of property, it has significance.

transition: (property name) | (duration) | (timing function) | (delay);

CSS Animation

CSS Animation: CSS Animations is a technique to change the appearance and behavior of various elements in web pages. It is used to control the elements by changing their motions or display.

It has two parts, one which contains the CSS properties which describe the animation of the elements and the other contains certain keyframes which indicate the animation properties of the element and the specific time intervals at which those have to occur.

The @keyframes rule: Keyframes are the foundations with the help of which CSS Animations works. They define the display of the animation at the respective stages of its whole duration.

Animation Properties: There are certain animation properties given below:

animation-name: It is used to specify the name of the @keyframes describing the animation.

animation-duration: It is used to specify the time duration it takes animation to complete one cycle.

animation-timing-function: Specifies how the animation makes transitions through keyframes. It can have the following values:

1. **ease:** The animation starts slowly, then fast, and then finally ends slowly (this is the default)
2. **linear:** The animation plays with the same speed from start to end
3. **ease-in:** The animation plays with a slow start
4. **ease-out:** The animation plays with a slow end
5. **ease-in-out:** The animation starts and ends slowly.

animation-delay: It is used to specify the delay when the animation starts.

animation-iteration-count: It is used to specify the number of times the animation will repeat. It can specify as infinite to repeat the animation indefinitely.

animation-direction: Specifies the direction of the animation. It can have the following values:

1. **normal:** The animation is played forward. This is the default value.
2. **reverse:** The animation is played in the reverse direction i.e. backward.
3. **alternate:** The animation is played forwards first, and then backward.
4. **alternate-reverse:** The animation is played backward first, and then forwards.

Classwork: Build an Image Gallery
Project: Basic Blog App