CSS Animations

1. **CSS Animations:**
    a. An animation makes an element change gradually from one style to another.
    b. You can add as many as properties you want to add. We can also specify the changes in percentage.
    c. 0% specify the start of the animation and 100% specify its completion.
    d. CSS animations make it possible to animate transitions from one CSS style configuration to another.

2. **What is animation exactly?**
    a. Animation is a method in which figures are manipulated to appear as moving images.

3. **Advantages of animations:**
    a. Easy to use simple animations
    b. It is not required to hover the element to see the changes by default it is automatically will start the changes.
    c. It performs smoothly under moderate system load.

4. **Animation Syntax:**
    a. **1st way:** when only the start and end is there.

b. **Syntax:**

```
@keyframes identifier {
                    from {
                        properties;
                    }
                    to {
                        properties;
                    }
}
```

c. **2nd way:** when you have to provide in stages or phases.

d. **Syntax:**

```
@keyframes identifier {
                    0% {
                        properties;
                    }
                    10% {
                        properties;
                    }
                    20% {
                        properties;
                    }
                    .
                    .
                    .
                    100% {
                        properties;
                    }
}
```

e. Here, the *identifier* is nothing but the name of the animation.
f. Also, The CSS ***@keyframes*** rule is used to control the steps in an animation sequence by defining CSS styles for points along the animation sequence.

5. *CSS* Animations properties:

a. animation-name
b. animation-duration
c. animation-timing-function
d. animation-delay
e. animation-iteration-count
f. animation-direction
g. animation-fill-mode
h. animation-play-state

6. animation-name:

a. Specifies the name of the *@keyframes* at-rule describing the animation's keyframes.
b. **Syntax**:

```
animation-name: name_of_the_animation (text);
```

7. animation-duration:

a. It configures the length of time that an animation should take to complete one cycle.
b. **Syntax**:

```
animation-duration: time (in s or ms)
```

8. **animation-timing-function:**
    a. Configures the timing of the animation; that is, how the animation transitions through keyframes, by establishing acceleration curves.
    b. **Syntax:**

```
animation-timing-function: linear => same speed throught the the transition effect
ease(Default) => slow start, then fast, end slowly
ease-in => slow start
ease-out => slow end
ease-in-out => slow start and end
```

9. **animation-delay:**
    a. The *animation-delay* property specifies a delay for the start of an animation.
    b. Defined in seconds (s) or milliseconds (ms), this value determines how long to wait before beginning the animation.
    c. Negative values start the animation as if it had already been playing for the specified duration.
    d. **Syntax:**

```
animation-delay: time (in s or ms);
```

10. **animation-iteration-count:**
    a. The *animation-iteration-count* property in CSS specifies the number of times an animation should be repeated.
    b. It can also be set to infinite to repeat the animation indefinitely.

c. **Syntax**:

```
animation-iteration-count: number | infinite;
```

11. **animation-direction:**
    a. The *animation-direction* CSS property sets whether an animation should play forward, backward, or alternate back and forth between playing the sequence forward and backward.
    b. It controls the visual flow and repetition behavior of animations in web pages.
    c. **Syntax**:

```
animation-direction: normal | reverse | alternate | alterna
te-reverse;
```

    d. *normal* ⇒ The animation is played as normal.
    e. *reverse* ⇒ The animation is played in reverse direction.
    f. *alternate* ⇒ The animation is played forward first, then backwards.
    g. *alternate-reverse* ⇒ The animation is played backwards then forward.

12. **animation-fill-mode:**
    a. The *animation-fill-mode* property in CSS is used to define the styles that an animation applies to an element before and after it executes.
    b. By default, CSS animations do not affect an element until the first keyframe is played or after the last keyframe is played.

c. **Syntax**:

```
animation-fill-mode: none | forwards | backwards | both;
```

d. *none* ⇒ Default value. The animation does not apply any styles to the element before or after its execution.

e. *forwards* ⇒ After the animation completes, the element retains the styles from the last keyframe.

f. *backwards* ⇒ The element applies the styles from the first keyframe before the animation starts.

g. *both* ⇒ This property is used to follow the rules for both forwards and backward.

13. **animation (Shorthand Property):**

a. *animation* is the shorthand property for all the different different properties.

b. **Syntax**:

```
animation: name duration timing-function delay iteration-count direction fill-mode;
```

c. Example ⇒

```
animation: rainbow 5s linear 2s infinite alternate forwards;
```

14. **animation-play-state:**
    a. The *animation-play-state* property in CSS controls animation playback: paused stops, running starts.
    b. **Syntax**:

```
animation-play-state: paused|running;
```

    c. The animation-play-state property is listed below ⇒
        i. *paused* ⇒ Specifies that the animation is paused.
        ii. *running* ⇒ The default value specifies that the animation is running.