CSS animations

CSS animations allow you to animate changes to CSS properties without requiring JavaScript. You can create keyframe-based animations that define how the properties should change over time. These animations can control multiple properties, provide looping, delays, and more complex effects.

Key Components of CSS Animations:

1. **@keyframes**: Defines the animation's behavior at specific moments (or "keyframes") during its duration.
2. **animation** property: Used to apply the animation to an element and control its timing, duration, iteration count, and other aspects.

Basic Structure:

1. **@keyframes** defines the steps of the animation.
2. **animation** or its shorthand applies the animation to an element.

Keyframes Syntax:

```
@keyframes animation-name {
  0% { property: value; }
  100% { property: value; }
}
```

- Keyframes define how properties change over time, using percentage values from `0%` (start) to `100%` (end).

Example:

Animate the background color from blue to red:

```
@keyframes color-change {
  0% {
    background-color: blue;
  }
  100% {
    background-color: red;
  }
}

.box {
    width: 100px;
    height: 100px;
    animation: color-change 3s linear infinite;
}
```

Explanation:

- `@keyframes color-change`: The animation is named "color-change" and defines two keyframes, `0%` and `100%`, where the background color goes from blue to red.
- `.box`: The animation is applied to the `.box` class.
    - `3s`: The animation lasts 3 seconds.
    - `linear`: The animation progresses at a constant speed.
    - `infinite`: The animation repeats indefinitely.

`animation` Shorthand:

You can define all the animation properties in a single `animation` declaration:

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

Common Properties:

- **animation-name**: Specifies the name of the `@keyframes` animation (e.g., `color-change`).
- **animation-duration**: Sets the length of the animation (e.g., `3s`, `2ms`).
- **animation-timing-function**: Defines how the animation progresses (e.g., `ease`, `linear`).
- **animation-delay**: Delays the animation (e.g., `1s` delay before it starts).
- **animation-iteration-count**: Sets how many times the animation runs (e.g., `infinite`, `1`, `3`).
- **animation-direction**: Specifies the direction of the animation (e.g., `normal`, `reverse`, `alternate`).
- **animation-fill-mode**: Defines the state of the element before and after the animation (e.g., `forwards`, `backwards`).

Example with Multiple Keyframes:

```
@keyframes move {
  0% { transform: translateX(0); }
  50% { transform: translateX(100px); }
  100% { transform: translateX(0); }
}

.box {
  width: 100px;
  height: 100px;
  background-color: lightgreen;
  animation: move 2s ease-in-out infinite;
}
```

- The `.box` will move 100px to the right and back to its original position over 2 seconds, repeating infinitely with an `ease-in-out` timing.

Why Use CSS Animations?

- **Performance**: CSS animations are often hardware-accelerated, improving performance.
- **No JavaScript required**: Animations are easier to create and manage directly in CSS.

- **Chaining**: CSS allows for chaining animations and transitions.

CSS animations are versatile and powerful, enabling you to create smooth, engaging effects on web pages.