

# CSS Animations Review

## CSS Animation Basics

- **Definition:** CSS animations allow you to create dynamic, visually engaging effects on web pages without the need for JavaScript or complex programming. They provide a way to smoothly transition elements between different styles over a specified duration.
- **The @keyframes Rule:** This rule defines the stages and styles of the animation. It specifies what styles the element should have at various points during the animation.

### Example Code

```
@keyframes slide-in {  
  
  0% {  
  
    transform: translateX(-100%);  
  
  }  
  
  100% {  
  
    transform: translateX(0);  
  
  }  
}
```

- **animation Property:** This is the shorthand property used to apply animations.
- **animation-name:** This specifies the name for the @keyframes rule to use.
- **animation-duration:** This sets how long the animation should take to complete.
- **animation-timing-function:** This defines how the animation progresses over time (such as ease, linear, ease-in-out).
- **animation-delay:** This specifies a delay before the animation starts.
- **animation-iteration-count:** This sets how many times the animation should repeat.
- **animation-direction:** This determines whether the animation should play forwards, backwards, or alternate.

- **animation-fill-mode:** This specifies how the element should be styled before and after the animation.
- **animation-play-state:** This allows you to pause and resume the animation.

### Accessibility and the prefers-reduced-motion Media Query

- **The prefers-reduced-motion Media Query:** One of the primary accessibility concerns with animations is that they can cause discomfort or even physical harm to some users. People with vestibular disorders or motion sensitivity may experience dizziness, nausea, or headaches when exposed to certain types of movement on screen. The prefers-reduced-motion media query allows web developers to detect if the user has requested minimal animations or motion effects at the system level.

#### Example Code

```
.animated-element {  
  transition: transform 0.3s ease-in-out;  
}
```

```
@media (prefers-reduced-motion: reduce) {  
  .animated-element {  
    transition: none;  
  }  
}
```