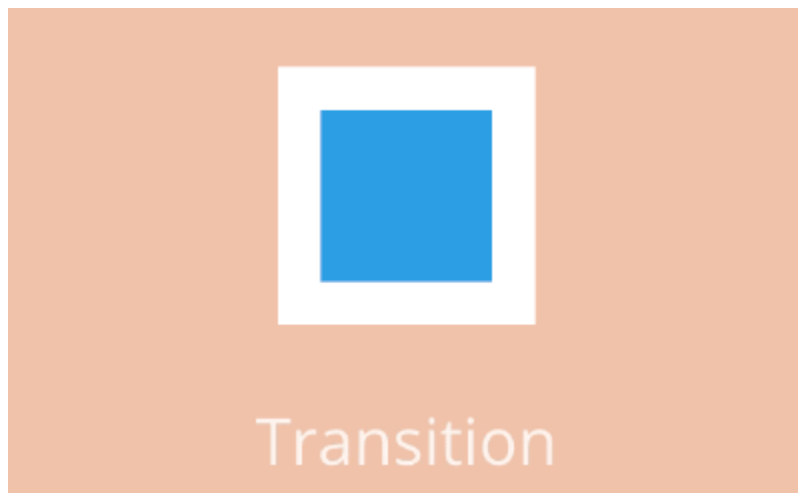


# CSS Animations & Transitions

CSS provides powerful tools to add **motion and interactivity** to web pages, helping create smooth user experiences without relying on heavy JavaScript. Animations and transitions bring life to interfaces, making interactions feel more natural, engaging, and responsive.



## Transitions

A **transition** in CSS allows properties to gradually change from one value to another when triggered by an event such as a hover, click, or focus. Instead of instantly jumping between styles, transitions create a smooth effect.

For example, hovering over a button can gradually change its background color:

```
button {  
  background-color: #2196f3;  
  transition: background-color 0.3s ease;  
}  
button:hover {  
  background-color: #0d47a1;  
}
```

### Key properties for transitions:

1. **transition-property**: Which CSS property to animate (e.g., color, width, all).
2. **transition-duration**: How long the transition lasts (e.g., 0.5s).
3. **transition-timing-function**: The speed curve (ease, linear, ease-in, ease-out, cubic-bezier).
4. **transition-delay**: When the transition starts after the trigger.

## Animations

*Animations* are more flexible than transitions because they allow continuous changes without needing user interaction. CSS animations rely on **keyframes**, which define stages of the animation.

Example of a simple animation:

```
@keyframes slideIn {  
  from { transform: translateX(-100%); }  
  to { transform: translateX(0); }  
}  
div {  
  animation: slideIn 1s ease-out;  
}
```

### Important animation properties:

1. **@keyframes**: Defines the sequence of changes (using from, to, or percentages like 0%, 50%, 100%).
2. **animation-name**: Refers to the keyframes.
3. **animation-duration**: Time the animation runs.
4. **animation-iteration-count**: Number of times it repeats (or infinite).
5. **animation-direction**: Controls playback (e.g., normal, reverse, alternate).
6. **animation-fill-mode**: Defines the state of the element before and after animation (forwards, backwards).

## **When to Use Animations & Transitions?**

Motion effects should enhance usability, not distract from it. Good uses include:

- Buttons that highlight smoothly when hovered or clicked.
- Navigation menus that slide or fade in.
- Loading indicators that reassure users.
- Highlighting form errors with subtle shakes or fades.

Poor use would be unnecessary flashing, overly long animations, or effects that slow down user interaction.