

CSS Advanced

Animations



- 1. CSS Animations features**
- 2. @keyframes**
- 3. Animation properties**
- 4. Timing functions**
- 5. Steps**
- 6. Iterations count and delays**
- 7. Direction, shorthand**
- 8. Events and transition**
- 9. Stoppping animations and events**

Section 1

CSS Animations features

1. CSS Animations features

- An animation lets an element gradually change from one style to another.
- CSS animations are made up of two basic building blocks:
 1. **Keyframes** - define the stages and styles of the animation.
 2. **Animation Properties** - assign the @keyframes to a specific CSS element and define how it is animated.
- **Demo:**
https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3

Section 2

@keyframes

2. @keyframes

- Keyframes are used to specify the values for the animating properties at various stages of the animation.
- Keyframes are specified using a specialized CSS at rule - *@keyframes*.
- Each *@keyframes* is composed of:
 - **Name of the animation:** A name that describes the animation, for example, `bounceIn`.
 - **Stages of the animation:** Each stage of the animation is represented as a percentage. 0% represents the beginning state of the animation. 100% represents the ending state of the animation. Multiple intermediate states can be added in between.
 - **CSS Properties:** The CSS properties defined for each stage of the animation timeline.

2. @keyframes

➤ Example:

```
@keyframes bounceIn {  
  0% {  
    transform: scale(0.1);  
    opacity: 0;  
  }  
  60% {  
    transform: scale(1.2);  
    opacity: 1;  
  }  
  100% {  
    transform: scale(1);  
  }  
}
```

This @keyframes has three stages.

1. (0%), the element is at opacity 0 and scaled down to 10 percent of its default size, using CSS transform scale.
2. (60%) the element fades in to full opacity and grows to 120 percent of its default size.
3. (100%), it scales down slightly and returns to its default size.

Section 3

Animation properties

3. Animation properties

- Once the @keyframes are defined, the **animation properties** must be added in order for your animation to function.
- Animation properties do two things:
 - They assign the @keyframes to the elements that you want to animate.
 - They define how it is animated.
- You must add the following two animation properties for the animation to take effect:
 - **animation-name**: The name of the animation, defined in the @keyframes.
 - **animation-duration**: The duration of the animation, in seconds

Section 4

Timing function

4. Function and transform order

- The **animation-timing-function**: Defines the speed curve or pace of the animation.
- The animation-timing-function property can have the following values:
 - *ease* - Specifies an animation with a slow start, then fast, then end slowly (this is default)
 - *linear* - Specifies an animation with the same speed from start to end
 - *ease-in* - Specifies an animation with a slow start
 - *ease-out* - Specifies an animation with a slow end
 - *ease-in-out* - Specifies an animation with a slow start and end
 - *cubic-bezier(n,n,n,n)* - Lets you define your own values in a cubic-bezier function

Section 5

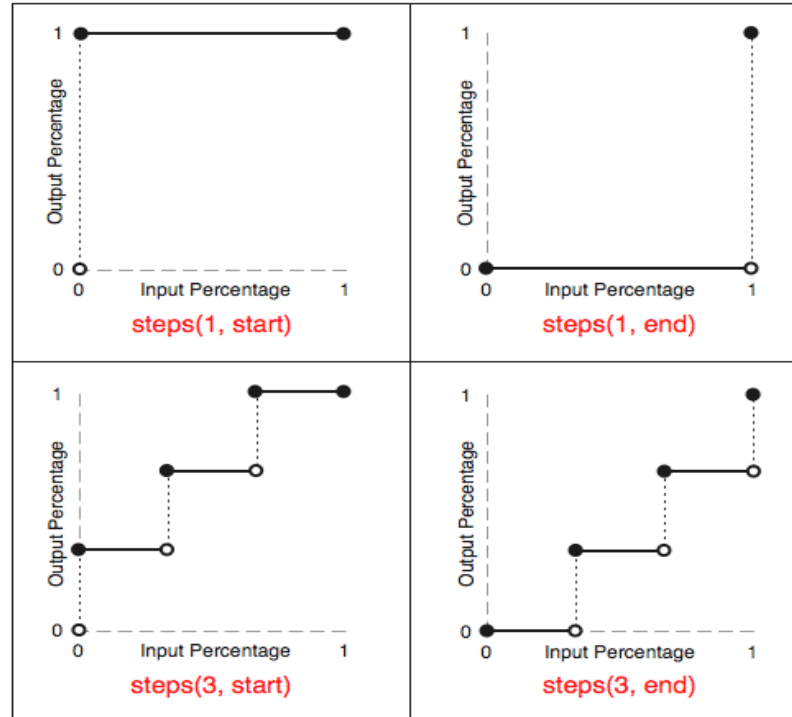
Steps

5. Steps

- Steps() is a timing function that allows us to break an animation or transition into segments, rather than one continuous transition from one state to another.
- The function takes two parameters
 - A positive integer which specifies the number of intervals in the function.
 - An optional value of either *start* or *end*, which specifies the point at which the change of values occur within the interval.
- **Syntax:** `steps(<integer>[, [start | end]]?)`

5. Steps

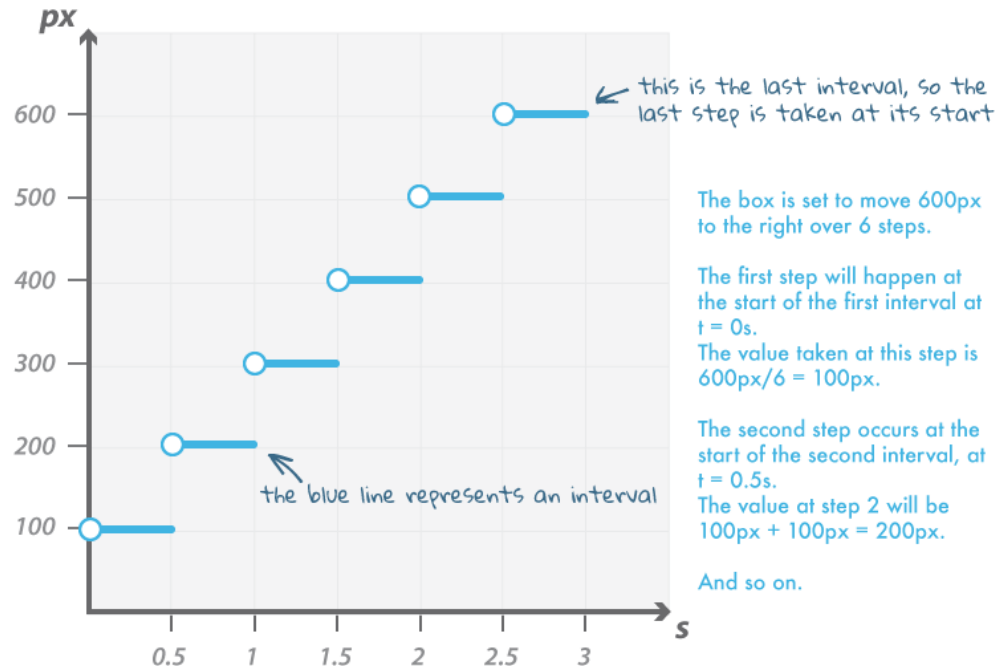
- Visual representation of examples of the steps() function



5. Steps

➤ Example

```
transition-duration: 3s;  
transition-timing-function: steps(6,start);
```



Section 6

Iterations count and delays

6. Iterations count and delays

- The **animation-iteration-count** property specifies the number of times an animation should be played.
- **Syntax:** *animation-iteration-count: number|infinite|initial|inherit;*
- **Demo:**
https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_animation-iteration-count2

6. Iterations count and delays

- The **animation-delay** property specifies a delay for the start of an animation.
- The animation-delay value is defined in seconds (s) or milliseconds (ms).
- **Syntax:** *animation-delay: time|initial|inherit;*
- **Demo:**
https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_animation-delay2

Section 7

Direction, shorthand

7. Direction, shorthand

- The **animation-direction** property defines whether an animation should be played forwards, backwards or in alternate cycles.
- **Syntax:** *animation-direction: normal|reverse|alternate|alternate-reverse|initial|inherit;*

7. Direction, shorthand

- The **animation** property is a shorthand property for:
 - animation-name
 - animation-duration
 - animation-timing-function
 - animation-delay
 - animation-iteration-count
 - animation-direction
 - animation-fill-mode
 - animation-play-state
- **Syntax:** *animation: name duration timing-function delay iteration-count direction fill-mode play-state;*

Thank you

