CSS TRANSITIONS AND ANIMATIONS

Css transitions

CSS transitions allow you to change property values smoothly, over a given duration. The transition CSS property is a shorthand property for t**ransition-property, transition-duration, transition-timing-function, and transition-delay.**

Transition properties

Transition: A shorthand property for setting the four transition properties into a single property

Transition-delay: Specifies a delay (in seconds) for the transition to take effect

Transition-duration: Specifies how many seconds or milliseconds a transition effect takes to complete

Transition-property: Specifies the name of the CSS property the transition effect is for

Transition-timing-function: Specifies the speed curve of the transition effect

USE OF TRANSITIONS

To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

Note: If the duration part is not specified, the transition will have no effect, because the default value is 0.

Example

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;/Shorthand*/
}
```

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition-property: width;
  transition-duration: 2s;
}
```

The transition effect will start when the specified CSS property (width) changes value.

```
div:hover {
  width: 300px;/*Notice that when the cursor mouses out of the element, it will gradually change back to
```

Syntax

```
/* Apply to 1 property */

/* property name | duration */

transition: margin-right 4s;

/* property name | duration | delay */

transition: margin-right 4s 1s;

/* property name | duration | timing function */

  transition: margin-right 4s ease-in-out;
```

```
/* property name | duration | timing
function | delay */

transition: margin-right 4s
ease-in-out 1s;

/* Apply to 2 properties */

transition: margin-right 4s, color
1s;

/* Apply to all changed properties */

transition: all 0.5s ease-out;
```

Rwanda Coding Academy

Specify the Speed Curve of the Transition

The `transition-timing-function` property specifies the speed curve of the transition effect. The transition-timing-function property can have the following values:

- `ease` - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
- `linear` - specifies a transition effect with the same speed from start to end
- `ease-in` - specifies a transition effect with a slow start
- `ease-out` - specifies a transition effect with a slow end
- `ease-in-out` - specifies a transition effect with a slow start and end

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}
div:hover {width: 300px;}
```

```
#div1
{transition-timing-function:
linear;}
#div2
{transition-timing-function:
ease;}
#div3
{transition-timing-function:
ease-in;}
#div4
{transition-timing-function:
ease-out;}
#div5
{transition-timing-function:
ease-in-out;}
```

Rwanda Coding Academy

Delay the Transition Effect

The `transition-delay` property specifies a delay (in seconds) for the transition effect.

The following example has a 1 second delay before starting:

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
  transition-delay: 2s;
}
```

Property to change

```
div:hover {
  width: 300px;
}
```

Change Several Property Values

```
div {
  transition: width 2s, height 2s, transform 2s ;
}
```

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s, height 2s,
transform 2s;
}

div:hover {
  width: 300px;
height:300px;
  transform: rotate(180deg);
}
```

Browser prefixes for older browsers

```
div {
  width: 100px;
  height: 100px;
  background: red;
  -webkit-transition: width 2s; /* Safari */
  transition: width 2s;
}
#div1 {-webkit-transition-timing-function:
linear;}
#div2 {-webkit-transition-timing-function:
ease;}

/* Standard syntax */
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
```

```
-webkit-transition: all 4s ease;/*SAFARI*/
-moz-transition: all 4s ease;/*MOZILLA*/
-ms-transition: all 4s ease;/*IE*/
-o-transition: all 4s ease;/*OPERA*/
transition: all 4s ease;
```

Rwanda Coding Academy

Transition Shorthand Property

```
div {
  transition-property: width;
  transition-duration: 450ms;
  transition-timing-function: linear;
  transition-delay: 1s;
}
```

```
div {
  transition: width 450ms linear 1s;
}
```

Rwanda Coding Academy

CSS ANIMATIONS

CSS animations make it possible to animate transitions from one CSS style configuration to another. Animations consist of two components, a style describing the CSS animation and a set of keyframes that indicate the start and end states of the animation's style, as well as possible intermediate waypoints.

There are key advantages to CSS animations over traditional script-driven animation techniques:

1. They're easy to use for simple animations; you can create them without even having to know JavaScript.
2. The animations run well.
3. Letting the browser control the animation sequence. It's always preferred to use the native browser abilities.
   Javascript[jQuery] animations are not real animations, they are faked, while the native CSS3 browser animations are in fact, animations.

ANIMATIONS PROPERTIES

Property	Description
@keyframes	Specifies the animation code
animation	A shorthand property for setting all the animation properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles
animation-duration	Specifies how long time an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies the name of the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing -function	Specifies the speed curve of the animation

Keyframes

- To use CSS animation, you must first specify some keyframes for the animation. Keyframes hold what styles the element will have at certain times.
- When you specify CSS styles inside the `@keyframes` rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element. See example below:

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: colorchange;
  animation-duration: 4s;
}
```

```
@keyframes colorchange {
  from {background-color: red;}
  to {background-color: yellow;}
}
```

- This example binds the "`colorchange`" animation to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow":
- The `animation-duration` property defines how long time an animation should take to

Keyframes-Example.2

- In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).It is also possible to use percent. By using percent, you can add as many style changes as you like.
- The following example will change the background-color of the element when the animation is 25%

```
/* The animation code */
@keyframes myanimationname {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: myanimationname;
  animation-duration: 4s;
}
```

```
……..
<body>
        <div></div>
</body>
</html>
```

Keyframes-Example3

The following example will change both the background-color, border-radius and the position of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

```
/* The animation code */
@keyframes bouncing {
  0%   {background-color:red; left:0px; top:0px; border-radius:0; }
  25%  {background-color:yellow; left:200px; top:0px;border-radius:50% 0 0 0;}
  50%  {background-color:blue; left:200px; top:200px;
border-radius:50% 50% 0 0;}
  75%  {background-color:green; left:0px; top:200px;border-radius:50% 50% 50% 0;}
  100% {background-color:red; left:0px; top:0px;border-radius:50% 50% 50% 50%}
   /* The element to apply the animation to
   */
   div {
     width: 100px;
     height: 100px;
     position: relative;
     background-color: red;
     animation-name: bouncing;
     animation-duration: 4s;
   }
```

```
.....
<body>
        <div></div>
</body>
</html>
```

Keyframes-Example 4

The following example will change the position of the h2 moving up and down several times through the animation cycle. The animation-delay:2s property specifies a delay of 2 seconds for the start of an animation. Negative values for animation-delay are also allowed.

```
/* Standard syntax */
 @keyframes bouncingHeading {
    0% {top: 0; }
    25% {top: 100px;     }
    50% {top: 0;     }
    75% {top: 100px;}
    100% {top: 0;}
}

    /* The element to apply the animation to */
    h2{
    color: blue;
    font-size:2em;
    animation-name:bouncingHeading;
    animation-duration:4s;
    animation-delay:2s;/*delay before start. If -2s the animation started 2s ago*/
    position:relative;
    }
```

```
.....
<body>
        <h2>Rwanda Coding Academy</h2>
</body>
</html>
```

Rwanda Coding Academy

Number of times the animation should repeat;

The `animation-iteration-count` property specifies the number of times an animation should run. you can specify `infinite` to repeat the animation indefinitely.

The following example will run the animation 3 times before it stops:

```
/* The animation code */
@keyframes bouncing {
  0%   {background-color:red; left:0px; top:0px; border-radius:0; }
  25%  {background-color:yellow; left:200px; top:0px;border-radius:50% 0 0 0;}
  50%  {background-color:blue; left:200px; top:200px;
border-radius:50% 50% 0 0;}
  75%  {background-color:green; left:0px; top:200px;border-radius:50% 50% 50% 0;}
  100% {background-color:red; left:0px; top:0px;border-radius:50% 50% 50% 50%}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: bouncing;
  animation-duration: 4s;
 animation-iteration-count: 3;/* YOu can use infinite*/
```

```
.....
<body>
        <div></div>
</body>
</html>
```

animation-direction

The `animation-direction` property specifies whether an animation should be played forwards, backwards or in alternate cycles.

The animation-direction property can have the following values:

- `normal` - The animation is played as normal (forwards). This is default
- `reverse` - The animation is played in reverse direction (backwards)
- `alternate` - The animation is played forwards first, then backwards
- `alternate-reverse` - The animation is played backwards first, then forwards

```
/* Example:Add animation-direction: alternate; to example
3The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: bouncing;
  animation-duration: 4s;
 animation-iteration-count: 3;/* YOu can use infinite*/
 animation-direction: alternate;/* YOu can use infinite*/
}
```

```
……..
<body>
        <div></div>
</body>
</html>
```

Specify the Speed Curve of the Animation

The `animation-timing-function` property specifies the speed curve 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

Timing function example

```
#div1 {animation-timing-function: linear;opacity: 0.6;}
#div2 {animation-timing-function: ease;opacity: 0.1;}
#div3 {animation-timing-function: ease-in;opacity: 0.5;}
#div4 {animation-timing-function: ease-out;opacity: 0.3;}
#div5 {animation-timing-function: ease-in-out;opacity:1;}
```

```
@keyframes mymove {
  from {left: 0px;}
  to {left: 300px;}
}
```

```
/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: mymove;
  animation-duration: 4s;

}
```

```
<div id="div1">linear</div>
<div id="div2">ease</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">ease-in-out</div>
```

Rwanda Coding Academy

Animation Shorthand Property

```
div {
  animation-name: example;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}
div {
  animation: example 5s linear 2s infinite alternate;
```