Full Stack Development

Week 4: Write a Program on Transitions and Animations in CSS

AIM:

Description:

4.1 Transitions:

CSS transitions allows you to change property values smoothly, over a given duration.

In this week we will learn about the following properties:

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

a) transition

The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

Code:

```
<!DOCTYPE html>
<html>
<style>
div {
        width: 100px;
        height: 100px;
        background: red;
        transition: width 2s, height 4s;
}

div:hover {
        width: 300px;
        height: 300px;
}
</style>
<body>

        <h1>The transition Property</h1>

        <p>Hover over the div element below, to see the transition effect:</p>

        <div></div>
```

```
</body>
</html>
```

Output:
Practice in lab

b)transition-delay

The transition-delay property specifies a delay (in seconds) for the transition effect.The following example has a 1 second delay before starting:

Code:

```
<!DOCTYPE html>
<html>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 3s;
    transition-delay: 1s;
}

div:hover {
    width: 300px;
}
</style>
<body>

    <h1>The transition-delay Property</h1>

    <p>Hover over the div element below, to see the transition effect:</p>

    <div></div>

    <p>
        <b>Note:</b> The transition effect has a 1 second delay before
        starting.
    </p>

</body>
</html>
```

Output:
Practice in lab

c)

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
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
        width: 100px;
        height: 100px;
        background: red;
        transition-property: width;
        transition-duration: 2s;
        transition-timing-function: linear;
        transition-delay: 1s;
}

div:hover {
        width: 300px;
}
</style>
</head>
<body>

        <h1>The transition Properties Specified One by One</h1>

        <p>Hover over the div element below, to see the transition effect:</p>

        <div></div>

        <p>
                <b>Note:</b> The transition effect has a 1 second delay before
        starting.
    </p>

</body>
</html>
```

Output:
Practice in lab

4.2 Animations

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

CSS Animation Properties

The following table lists the @keyframes rule and all the CSS animation 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

a) The @keyframes Rule

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.

The following example binds the "example" animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
        width: 100px;
        height: 100px;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
}

@keyframes example {
        from {background-color: red;
}

to {
        background-color: yellow;
}
}
</style>
</head>
<body>

        <h1>CSS Animation</h1>
        <div></div>

        <p>
                <b>Note:</b> When an animation is finished, it goes back to its
                original style.
        </p>

</body>
</html>
```

Output:
Practice in lab

Note: The animation-duration property defines how long an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

b)

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 <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```

Output:
Practice in lab

c)

Delay an Animation

The animation-delay property specifies a delay for the start of an animation.

The following example has a 2 seconds delay before starting the animation:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-delay: 2s;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-delay property specifies a delay for the start of an animation. The following example has a 2 seconds delay before starting the animation:</p>

<div></div>

</body>
</html>
```

Output:
Practice in lab

d)

Set How Many Times an Animation Should Run

The animation-iteration-count property specifies the number of times an animation should run.

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-iteration-count property specifies the number of times an animation should run. The following example will run the animation 3 times before it stops:</p>

<div></div>

</body>
</html>
```

Output:
Practice in lab

e)

Run Animation in Reverse Direction or Alternate Cycles

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

The following example will run the animation in reverse direction (backwards):

Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-direction: reverse;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-direction property specifies whether an animation should be played forwards,
backwards or in alternate cycles. The following example will run the animation in reverse
direction (backwards):</p>

<div></div>

</body>
</html>
```

Output:
Practice in lab

Result: Thus, in the above programs successfully executed without errors
Using CSS3 transitions and animations   eclipse editor**.**