CSS Animations:

What are CSS 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.

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.

Example

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

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

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).

Delay an Animation

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

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:

Example

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}
```

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

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

Animation Example: 1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <style>
        body
        {
            margin: 0;
            background-color: cadetblue;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;

        }
        .container
        {
            width: 300px;
            height: 300px;
            border: 2px solid white;

        }
        .box
        {
            width: 100px;
            height: 100px;
            background-color: white;
            animation-name: animation;
            animation-duration: 5s;
            animation-iteration-count: infinite;
            animation-timing-function: ease-out;
            animation-direction: alternate-reverse;
        }
```

```
        @keyframes animation
        {
            0%, 100%
            {
                transform: translate(0,0);
                /* border-radius: 50%;
                transform: rotate(360deg); */
                background-color: white;
            }
            25%
            {
                transform: translate(200px, 0);
                background-color: brown;
                /* border-radius: 50%;
                transform: rotate(360deg); */
            }
            50%
            {
                transform: translate(200px, 200px);
                background-color: coral;
                /* border-radius: 50%;
                transform: rotate(360deg); */
            }
            75%
            {
                transform: translate(0,200px);
                background-color: black;
            }
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">

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

Animation Example: 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=<device-width>, initial-
scale=1.0">
    <title>Document</title>

    <style>
        .container
        {
            width: 200px;
            height: 200px;
            background-color: red;
            animation-name: animation;
            animation-duration: 4s;
            animation-delay: 2s;
            animation-iteration-count: 3;
        }
        @keyframes animation
        {
            from
            {
                background-color: yellow;
            }
            to
            {
                background-color: brown;
            }
        }
    </style>
</head>
<body>
    <div class="container">

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

Animation Example: 3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <style>
        body
        {
            margin: 0;
            background-color: cadetblue;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;

        }
        .loader
        {
            display: flex;
            align-items: center;
            justify-content: center;
            gap: 10px;
        }
       .dot
       {
        width: 10px;
        height: 10px;
        border-radius: 50%;
        background-color: white;
        animation:animation 0.8s alternate infinite ease;
       }
       h2
       {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 35px;
        font-weight: bold;
```

```
        color: white;
       }
       @keyframes animation
       {
        form
        {
            transform: translate(-10px);
        }
        to
        {
            transform: translateY(10px);
        }
       }
       .dot:nth-child(2)
       {
        animation-delay: 0.1s;
       }
       .dot:nth-child(3)
       {
        animation-delay: 0.2s;
       }
       .dot:nth-child(4)
       {
        animation-delay: 0.3s;
       }
    </style>
</head>
<body>
    <div class="loader">
        <h2>Loading</h2>
        <div class="dot"></div>
        <div class="dot"></div>
        <div class="dot"></div>
        <div class="dot"></div>
    </div>
</body>
</html>
```