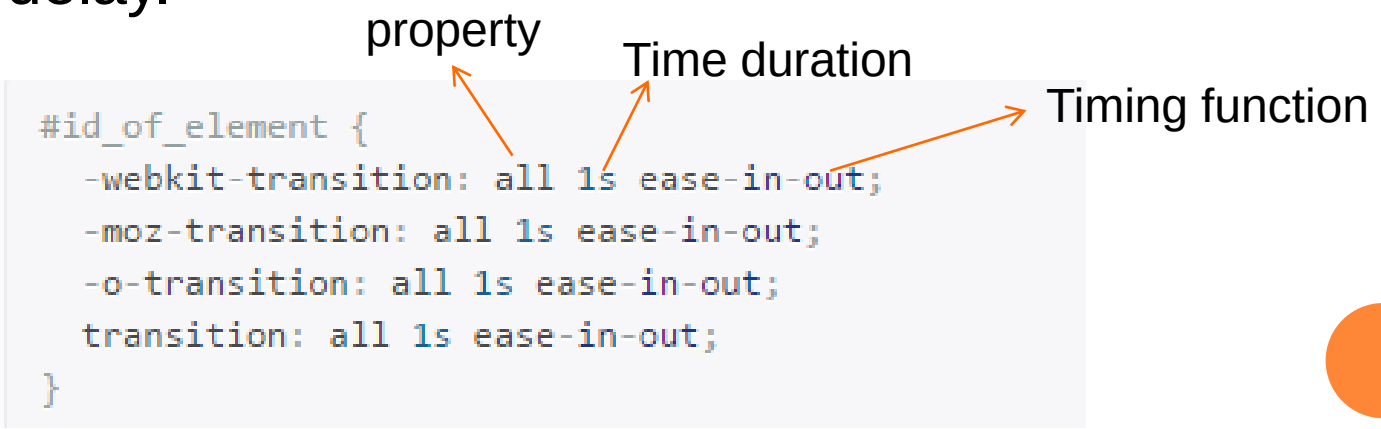
Umadevi Balakrishnan

CSS CORP
Wired to SUPPORT

Transition Animation

Transition

- Adding an effect when changing from one style to another
- without using Flash or JavaScript
- Transition rule will tell the browser to animate all/specific properties attached to the object for specific seconds using a timing function , without delay.



Timing

- 6 types of transition-timing-functions:
- **linear:** the transition will have constant speed from start to end.
- **ease:** the transition will start slowly, then get faster, before ending slowly.
- **ease-in:** the transition will start slowly.
- **ease-out:** the transition will end slowly.
- **ease-in-out:** the transition starts and ends slowly.
- **cubic-bezier:** define specific values for your own transition.

```
div
{


    background-image: url("plane.jpg");
    background-repeat: no-repeat;
    width:500px;
    height: 520px;
    -webkit-transform: translate(10px,500px);
    -webkit-transition:all 2s ease-in-out;

}
div:hover
{
    -webkit-transform: translate(450px,-350px) scale(2);

}
```

Samples

- Sample1 – Plane
- Sample2 – Car1
- Sample3- Car2
- Sample4 – Pencil
- Sample 5 - Bird
- Sample6 -Rocket
- Sample7 -Card Flip

Animation

Animation???

- CSS3 animations can replace animations created by Flash and JavaScript in existing web pages
- An animation lets an element gradually change from one style to another.
- Can change as many properties as many times wanted
- Can specify when the change will happen in percent, or using the keywords "from" and "to" (which represents 0% and 100%).
- 0% represents the start of the animation, 100% is when the animation is complete.

Browser compatibility

Property					
@keyframes	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-
animation	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

@Keyframes

- Rule for creating animation
- Specify a CSS style inside the @keyframes rule and the animation will gradually change from the current style to the new style

```
1   div
2   {
3       width: 100px;
4       height: 100px;
5       background: red;
6       -webkit-animation: mysecond 2s;
7   }
8
9   @-webkit-keyframes myfirst
10  {
11      from {background: red;}
12      to{background: yellow;}
13
14  }
15
16  @-webkit-keyframes mysecond {
17      0%   {background: red;}
18      25%  {background: yellow;}
19      50%  {background: blue;}
20      100% {background: green;}
21  }
```

Animation

- When an animation is created in the @keyframe rule, must bind it to a selector, otherwise the animation will have no effect.
- Bind the animation to a selector (element) by specifying at least these two properties:
  - the name of the animation
  - the duration of the animation

```
div
{
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    -webkit-animation-name: mythird 5s;
}

@-webkit-keyframes mythird {
    0%   {background: red; left:0px; top:0px;}
    25%  {background: yellow; left:200px; top:0px;}
    50%  {background: blue; left:200px; top:200px;}
    75%  {background: green; left:0px; top:200px;}
    100% {background: pink; left:0px; top:0px;}
}
```

Sample

```
div
{
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    -webkit-animation-name: mythird;
    -webkit-animation-duration: 5s;
    -webkit-animation-timing-function: linear;
    -webkit-animation-delay: 2s;
    -webkit-animation-iteration-count: infinite;
    -webkit-animation-direction: alternate;
    -webkit-animation-play-state: running;
}
@-webkit-keyframes mythird {
    0%   {background: red; left:0px; top:0px;}
    25%  {background: yellow; left:200px; top:0px;}
    50%  {background: blue; left:200px; top:200px;}
    75%  {background: green; left:0px; top:200px;}
    100% {background: pink; left:0px; top:0px;}
}
```

```
div
{
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
-webkit-animation: mythird 5s linear 2s infinite alternate;
}
@-webkit-keyframes mythird {
    0%   {background: red; left:0px; top:0px;}
    25%  {background: yellow; left:200px; top:0px;}
    50%  {background: blue; left:200px; top:200px;}
    75%  {background: green; left:0px; top:200px;}
    100% {background: pink; left:0px; top:0px;}
}
```

Sprite Sheet Animation

```css
#ball
{
    background: url("bounce_ball.jpg");
    width: 30px;
    height: 40px;
    -webkit-transform: scale(2);
   -webkit-animation: bounce_ball 0.7s steps(6) infinite;
}

#hi
{
    background: url("hi.png");
    width: 50px;
    height: 72px;
    -webkit-animation: hi 0.8s steps(10) infinite;


    }

@-webkit-keyframes bounce_ball
{
    from{background-position: 0px;}
    to{background-position: -240px;}


}

@-webkit-keyframes hi
{
    from{background-position: 0px;}
    to{background-position: -500px;}


}
```

```html
<!DOCTYPE html>

<html>
<head>
    <link rel="stylesheet" href="style4.css" type="text/css">
    <title>Page Title</title>
</head>

<body>
<div id="ball"></div>
<br><br>
<div id="hi"></div>

</body>
</html>
```

Sample

More Animations

- Sample 1

References

- www.W3schools.com
- www.codecademy.com
- www.adamkhoury.com
- http://davidwalsh.name/css-flip
- http://24ways.org/2010/intro-to-css-3d-transforms/
- www.zengarden.com