CSS3

TRANSITIONS

Normally when the value of a CSS property changes, the results are **instant**.

Transitions are used to animate smoothly from the one state to another state **over time**.

For example, if a background-color changes on hover, we can **make it gradually fade from one background-color to the other**, instead of a sudden change.

Transitions are basically telling the browser **two key things**:

1. Which properties will be animated? This is defined using the transition-property property.

2. How long will the transition take? This is defined using the transition-duration property.

There are also two **optional extras**:

3. Will there be a delay?

This is defined using the transition-delay property.

4. Will the speed change during transition?

This is defined using the transition-timing-function property.

The steps

Here are the **three steps** to create a simple transition using only CSS3.

Step 1: Declare the **original state of the element**.

```css
a
{
    color: white;
    background-color: red;
}
```

2. Declare the **final state** of the element.

```css
a
{
    color: white;
    background-color: red;
}

a:hover
{
    color: black;
    background-color: yellow;
}
```

3. Include the **transition functions** in the default declaration.

```css
a
{
    color: white;
    background-color: red;
    transition: background-color 1s;
}

a:hover
{
    color: black;
    background-color: yellow;
}
```

The important point to note is that the **transition is declared in the default declaration**, not the final declaration.

transition-property

The transition-property **lists the CSS properties** that will be changed during the animation.

```
a
{
    transition-property: background-color;
}
```

Some properties that can be transitioned, and others cannot. Properties that can be made to transition include **background, border, and box model properties**.

A full list of all **properties that are able to be given transition** is listed here:

http://dev.w3.org/csswg/css3-transitions/#properties-from-css-

We can provide any **number of CSS properties**, separated by commas.

```css
a
{
    transition-property:
        background-color, color, padding;
}
```

Alternatively, we can use the keyword "all" to indicate that **every supported property should be animated**.

```css
a
{
    transition-property: all;
}
```

transition-duration

The transition-duration property determines **how long the transition will take**.

```css
a { transition-duration: 1s; }
```

Time is specified with **seconds** (s) or **milliseconds** (ms).

```css
a { transition-duration: 1s; }
a { transition-duration: 1000ms; }
```

Negative values are **treated as "0"** - which means the transition would be instant.

```css
a { transition-duration: -1s; }
a { transition-duration: 0; }
```

We can also provide **durations for multiple properties**, separated by commas.

```css
a { transition-duration: 1s, 300ms; }
```

transition-
timing-function

The transition-timing-function allows us to **control the pace of the transition**.

```
a
{
    transition-timing-function: linear;
    transition-timing-function: ease;
    transition-timing-function: ease-in;
    transition-timing-function: ease-out;
    transition-timing-function: ease-in-out;
    transition-timing-function:
        cubic-bezier(n,n,n,n);
}
```

linear	Smoothly over whole animation
ease	begins quickly then slows
ease-in	Speeds up
ease-out	Slows down
ease-in-out	Speeds up then slows down
cubic-bezier	Allows detailed timing functions

We can also provide **timing functions for multiple properties**, separated by commas.

```
a
{
    transition-timing-function:
        ease, linear, ease-out;
}
```

transition-delay

The transition-delay allows us to **delay the start of the transition**.

```css
a { transition-delay: 1s; }
```

Time is specified with **seconds** (s) or **milliseconds** (ms).

```
a { transition-delay: 1s; }
a { transition-delay: 1000ms; }
```

Negative values make the transition start **part of the way through the transition**.

```
a { transition-delay: -1s; }
```

We can also provide **delays for multiple properties**, separated by commas.

```
a { transition-delay: 1s, 300ms; }
```

Multiple
longhand values

We can combine **multiple values** for all of these properties.

```css
a
{
    transition-property:        color, margin;
    transition-duration:        1s,    1.5s;
    transition-timing-function: ease,  ease-in;
    transition-delay:           1s,    1s;
}
```

Shortand

The shorthand transition property allows us to **define all four values in one declaration**.

```
a
{
    transition:
        background-color 1s ease .5s;
}
```

The values **must be placed in the following order**:

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

However, you **do not need to specify** all values.

```css
a
{
    transition:
        background-color ease;
}
```

Multiple shorthand values

The shorthand property also allows us to add **multiple transitions**, separated by commas.

```css
a
{
    transition:
        background-color 1s ease .5s,
        color 1s linear .5s,
        padding-top 1s ease-out .5s;
}
```

Browser support

CSS3 Transitions ▤ - WD

Simple method of animating certain properties of an element, with ability to define property, duration, delay and timing function.

Current aligned   Usage relative   Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
								4.1	
8			43					4.3	
9		40	44					4.4	
10		41	45	8		8.4		4.4.4	
11	**12**	**42**	**46**	**9**	**32**	**9.1**	**8**	**44**	**46**
	13	43	47		33				
		44	48		34				
		45	49						

Russ Weakley

Max Design

Site: maxdesign.com.au

Twitter: twitter.com/russmaxdesign

Slideshare: slideshare.net/maxdesign

Linkedin: linkedin.com/in/russweakley