

CSS: test on browser support to transitions

When it comes to CSS transitions, our initial enthusiasm for this new feature should be followed by a more pragmatic analysis of what CSS transitions can actually do. After running a simple test on transitions and CSS properties, I came to the conclusion that browsers have still a long way to go before their support to CSS transitions can be safely defined as optimal. Moreover, CSS specifications allow only a small subset of properties to be affected by transitions, thus reducing their impact only to some little and fancy visual decorations on HTML elements. For that reasons, I'd like to discuss with you some interesting aspects of CSS transitions that I've noticed after my test.

The first thing that comes to our attention is the limited number of CSS properties that can be affected by a transition. For example, here's the complete code of my test:

```
.btn {
  display: inline-block;
  padding: 4px 10px 4px;
  margin-bottom: 0;
  font-size: 13px;
  line-height: 18px;
  color: #333333;
  text-align: center;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  vertical-align: middle;
  background-color: #f5f5f5;
  background-image: -moz-linear-gradient(top, #ffffff,
#e6e6e6);
  background-image: -ms-linear-gradient(top, #ffffff,
```

```
#e6e6e6);
    background-image: -webkit-gradient(linear, 0 0, 0
100%, from(#ffffff), to(#e6e6e6));
    background-image: -webkit-linear-gradient(top,
#ffffff, #e6e6e6);
    background-image: -o-linear-gradient(top, #ffffff,
#e6e6e6);
    background-image: linear-gradient(top, #ffffff,
#e6e6e6);
    background-repeat: repeat-x;
    border-color: #e6e6e6 #e6e6e6 #bfbfbf;
    border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1)
rgba(0, 0, 0, 0.25);
    border: 1px solid #cccccc;
    border-bottom-color: #b3b3b3;
    border-radius: 4px;
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0
1px 2px rgba(0, 0, 0, 0.05);
    cursor: pointer;
    -moz-transition: all 700ms ease-out;
    -webkit-transition: all 700ms ease-out;
    -o-transition: all 700ms ease-out;
    -ms-transition: all 700ms ease-out;
    transition: all 700ms ease-out;
    width: auto;

}
```

```
.btn:hover {
    width: 100px; /* works oddly due to the lack of a
stated width on the normal state */
    padding: 10px 20px; /* works */
    margin: 2em; /* works */
    border-width: 2px; /* not supported */
```

```
text-align: right; /* not supported */  
font-size: 18px; /* works */  
line-height: 22px; /* works */  
display: block; /* stops everything in Chrome */  
}
```

An interesting thing is on the last rule: `display: block` seems to entirely stop all effects in Chrome. Opera shows some peculiar problems with the font size augmented by the transition. All browsers agree when it comes to the lack of a stated width on the normal state: they simply make the element disappear at first and show it again with a curious bouncing effect.

Surely I need more tests to draw a definitive conclusion, but this first one seems to confirm all my doubts about the current browser support to CSS transitions.