Web Technologies — Week 4

Mikheil Rukhaia

International Black Sea University,
Tbilisi, Georgia

October 29, 2015

Outline

New CSS3 Features

Introduction

- CSS3 is a modular specification of CSS that adds dozens of new features.
- Modular means that specification is divided into modules, specifying a concrete feature.
- It also means that some modules are much further along than others for both the specification process and browser implementation.

Browser prefixes

- Not yet standardized modules are implemented in different manner in different browsers.
- Thus, each browser has its own prefix to indicate proprietary solution.
- For developers this means that newer CSS3 features should be listed for each browser using their prefixes.

Browser prefixes (ctd.)

Prefix	Browsers
`-ms-`	Internet Explorer
`-moz-`	Firefox, Camino, Seamonkey
`-o-`	Opera, Opera Mini, Opera Mobile
`-webkit-`	Safari, Chrome, Android, Silk, Blackberry, WebOS, etc.
`-khtml-`	Konqueror

Transparency

- CSS3 allows to specify transparency of an element via `opacity` property.
- Values of `opacity` can be a range from 0 to 1.
- 0 means that element is hidden (but it still participates in layout), 1 means completely opaque.
- `opacity` applies to any sub-elements as well.

Transparency (ctd.)

- It is possible to target transparency to one specific component of the element (background, foreground, border, etc.)
- You can set transparency using `rgba` color value (RGB plus alpha value).
- Example:

```
body {
  background-color: rgba(0,255,0,0.5);
  color: rgba(0,0,0,0.7);
}
```

Rounded corners

- Getting rounded corners is very easy via CSS3.
- You should specify `border-radius` property.
- Values can be set in any measure units, even in percentages.
- Example: test on different elements

```
border-radius: 1%;
border-radius: 1% 2%;
border-radius: 1% 2% 3% 4%;

border-radius: 1% / 2%;
border-radius: 1% 2% / 3% 4%;
border-radius: 1% 2% 3% 4% / 5% 6% 7% 8%;
```

Text shadow

- Shadow is important to make effect of a 3D, but be careful, it reduces performance.
- `text-shadow` makes a shadow for the text depending on several parameters.
- The horizontal offset, vertical offset, blur radius (optional) and color must be specified.
- Example:

```
text-shadow: .1em .2em gray;
text-shadow: -.1em -.2em gray;
text-shadow: .1em .2em .1em gray;
```

Text shadow (ctd.)

- It is possible to specify several shadow levels as well.
- Example:

```
h1 {
   width: 200px;
   background-color: green;
   color: black;
   text-align: center;
   text-shadow: -.05em -.05em .05em white,
                .03em .03em .05em gray;
}
```

Box shadow

- To add shadow to the box of an element `box-shadow` property is used.
- It is similar to `text-shadow`, but can have additional spread parameter.
- It is possible to render shadow inside the element box by adding `inset` keyword.
- Example:

```
box-shadow: inset 1em 2em 3em 4em gray;
```

CSS3 Transforms and Transition

Transitions

- Transitions change a state of the element smoothly.
- Transition is a recent feature, so browser prefixes are required.
- There are several parameters which can be configured for transition: `transition-property`, `transition-duration`, `transition-delay`, and `transition-timing-function`.
- Transition needs some event to be triggered (e.g. `:hover`, `:focus`, `:active`, etc.).

Transitions (ctd.)

- `transition-property` can be nearly any property that can be changed.
- There is even `all` keyword to apply transition to all properties together.
- `transition-duration` sets the amount of time for the animation either in seconds (`s`) or milliseconds (`ms`).
- `transition-delay` delays the start of the animation by a specified amount of time.

Transitions (ctd.)

- `transition-timing-function` can have one of the following values (default is `ease`):
  - `ease` starts slowly, accelerates quickly, then slows down at the end.
  - `linear` stays consistent from the beginning to end.
  - `ease-in` starts slowly, then speeds up.
  - `ease-out` starts fast, then slows down.
- In most cases `ease` or `linear` is perfectly OK, but there are some advanced speed controlling capabilities as well (used in animations).

Transitions (ctd.)

- Example:

```
a {
  transition-property: background-color,
                       opacity;
  transition-duration: 3s;
}

a:hover, a:focus {
  background-color: red;
}

a:active {
  opacity: 0;
  transition-delay: 5s;
}
```

Transitions (ctd.)

- You can specify values in the shorthand `transition` property as well.
- Example:

```
transition: all 3s linear;

transition: background-color 3s,
            opacity 1s 5s;
```

Transforms

- Transforms module makes it possible to rotate, relocate, resize, and skew HTML elements.
- Transformation is done via `transform` property.
- It can be applied to elements with:
  - replaced content (e.g. `img`, `canvas`, `form`, `input`, etc.)
  - `display: block;`
  - `display: inline-block;`
  - `display: inline-table;` (or any of the `table-*` display types)

Transforms (ctd.)

- To rotate an element, use the `rotate` function, where angle is specified in positive or negative `degs`.
- By default element rotates around its center, but this can be changed using `transform-origin` property.
- The latter takes two values for horizontal and vertical offsets.
- Example:

```
img {
  transform: rotate(-10deg);
  transform-origin: top left;
}
```

Transforms (ctd.)

- You can relocate an element using `translateX`, `translateY`, and `translate` functions.
- Analogously, there is `scaleX`, `scaleY`, and `scale` functions to resize an element.
- Example:

```
transform: translateX(50px);
transform: translateY(-25px);
transform: translate(-10px, -20%);

transform: scaleX(0.5);
transform: scaleY(1.5);
transform: scale(0.5, 2);
transform: scale(2);
```

Transforms (ctd.)

- Function `skewX`, `skewY`, and `skew` change the angle of the horizontal and/or vertical axis by the specified `degs`.
- Finally, you can combine these functions together in one `transform`.
- Example:

```
transform: skewX(10deg);
transform: skewY(20deg);
transform: skew(-10deg,-20deg);

transform: scale(1.5) rotate(-20deg);
```

Animations

- Using `@keyframes` you can make explicit animations.
- Making CSS animations is quite complex, so here is just a starting point.
- If you are interested in this topic, consult `http://www.anthonycalzadilla.com/`
- General syntax is the following:

```
@keyframes animation-name {
  keyframe { property: value; }
  keyframe { property: value; }
}
```

Animations (ctd.)

- After keyframes are defined, you need to construct an `animation`.
- It contains parameters similar to transitions, and additionally `animation-iteration-count`, `animation-direction`, etc.
- `animation-iteration-count` can have either number or `infinite` as a value.
- `animation-direction` can have `forward`, `reverse`, and `alternate` as a value.

Animations (ctd.)

- Example:

```
@keyframes moving {
  0% { left: 0px; }
  20% { left: 20px; }
  40% { left: 40px; }
  60% { left: 60px; }
  80% { left: 80px; }
  100% { left: 100px; }
}
div {
  position: relative;
  top: 0px; left: 0px;
  background-color: green;
  width: 100px; height: 100px;
  animation: moving 3s linear infinite
                                  alternate; }
```

Laboratory Work

Exercises

- Implement a menu shown in the picture.

- Combine transformation and transition to move images when hovering.
- Make a small animation.

Discussion?!