

Лабораторная работа № 10. Анимация. Создание выпадающего меню

Краткая теория

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

Объявление стиля состоит из двух частей: селектора и объявления. В HTML имена элементов нечувствительны к регистру, поэтому «h1» работает так же, как и «H1». Объявление состоит из двух частей: имя свойства (например, color) и значение свойства (grey). Селектор сообщает браузеру, какой именно элемент форматировать, а в блоке объявления (код в фигурных скобках) перечисляются формирующие команды — свойства и их значения.

Анимация

Свойства анимации поддерживаются браузерами IE10+, Firefox и Opera. Для браузеров Chrome и Safari перед свойством требуется добавить префикс -webkit.

Для создания анимации в CSS3 используется свойство @keyframes. Оно представляет собой контейнер, в который помещаются различные свойства оформления.

CSS keyframes позволяет производить изменения постоянно и автоматически, а не только в ответ на события мыши, как в transition. С помощью кейфреймов можно менять CSS стили для заданного селектора в любой точке в момент смены состояния или события, определенного через JQuery (например, скролинг). Данное правило используется в паре со свойством animation, с помощью которого можно задать duration, timing function, delay и direction. Свойства типа transform, если они входят в анимацию, объявляются внутри правила @keyframes.

Для начала каждое правило @keyframe должно иметь уникальное имя:

```
@keyframes animation-name{ }
```

Это имя используется в стилях элемента в свойстве animation:

```
.element {  
    animation: animation-name;  
}
```

В кейфрейме задаются правила, выраженные в процентах. Проценты представляют собой точки вдоль временной линии анимации, внутри которых задаются стили для рендеринга на элементе:

```
@keyframes animation-name {
  0% {
    color: pink;
  }
  50% {
    color: yellow;
  }
  100% {
    color: blue;
  }
}
```

В этом примере элемент `div` с классом `element` меняет цвет с розового на желтый и в конце на синий. Если точек всего две, можно использовать `from` и `to`:

```
@keyframes animation-name {
  from {
    color: black;
  }
  to {
    color: white;
  }
}
```

Свойство `animation`

Сам по себе `@keyframes` без инструкций, которые определяют длительность, временную функцию и направление, ничего не делает. Все эти инструкции работают точно так же, как и в свойстве `transition`. Чтоб анимировать элемент, надо добавить ему свойство `animation` и указать в нем имя анимации (1-е значение) и время (2-е значение), в течении которого она будет выполняться. Также можно устанавливать количество повторов анимации (3-е значение).

```
@keyframes имяАнимации {from {CSS свойства}}
```

```
/* Оформление элемента перед началом анимации */to {CSS свойства}/*  
Оформление элемента после завершения анимации *//#wrap1  
{animation:anim 4s 3;}
```

С помощью % можно более точно контролировать ход выполнения анимации, например, можно указать, что определенный элемент в начале анимации (0%) имеет один цвет, к середине (50%) окрашивается в другой цвет, а к концу (100%) – в третий цвет.

Префикс -webkit-

Для определения стилей элемента используется префикс -webkit в браузерах. На текущий момент, если вы делаете кроссбраузерную верстку, вам обязательно придется использовать -webkit префикс для правильного отображения стилей в некоторых браузерах (Safari и Chrome)

Приложения, основанные на WebKit или Blink, , поддерживают ряд специальных расширений WebKit для CSS.

Пример 1. Передвижение квадрата:

```
<html>  
<style type='text/css'>  
@-webkit-keyframes anim {  
0% {margin-left:3px;margin-top:3px;background-color:#7F0055;}  
30% {margin-left:3px;margin-top:250px;background-color:#7F0055;}  
60% {margin-left:500px;margin-top:250px;background-color:black;}  
100% {margin-left:3px;margin-top:3px;background-color:#7F0055;}  
}  
#wrap1 {  
border:2px #000 solid;  
background-color:#7F0055;  
height:100px;  
width:100px;  
font-size:2em;  
animation:anim 6s 3;  
-webkit-animation:anim 6s 3;  
}
```

```
</style>
<body>
<div id="wrap1"></div>
</body>
</html>
```

Пример 2 Куб с фото



```
<style type="text/css">
#holder {
margin:50px auto;
-webkit-perspective:600px;
-moz-perspective:600px;
text-indent: 0px;
}
.box {
position: relative;
margin: 0 auto;
height: 200px;
width: 200px;
-webkit-animation: spin 20s infinite linear;
-moz-animation: spin 20s infinite linear;
-webkit-transform-style: preserve-3d;
-moz-transform-style: preserve-3d;
}
.box div {
position: absolute;
height: 200px;
width: 200px;
box-shadow: 0px 0px 1px rgba(0, 0, 0, 0.5); -webkit-animation:rotate 15s
infinite linear;
-moz-animation:rotate 15s infinite linear;
}
```

```
.box div img {width:200px; height:200px;
border-radius:200px;
-webkit-animation:rot 10s infinite linear;
-moz-animation:rot 10s infinite linear;
padding: 0;
}
.box div.side1 {
background:rgba( 213, 62, 7, 0.8);
-webkit-transform:translateZ(100px);
-moz-transform:translateZ(100px);
}
.box div.side2 {
background:rgba( 127, 255, 212, 0.8);
-webkit-transform:rotateY(90deg) translateZ(100px);
-moz-transform:rotateY(90deg) translateZ(100px);
}
.box div.side3 {
background:rgba( 255, 219, 88, 0.8);
-webkit-transform:rotateY(180deg) translateZ(100px);
-moz-transform:rotateY(180deg) translateZ(100px);
}
.box div.side4 {
background:rgba( 153, 17, 153, 0.8);
-webkit-transform:rotateY(-90deg) translateZ(100px);
-moz-transform:rotateY(-90deg) translateZ(100px);
}
.box div.side5 {
background:rgba( 0, 71, 171, 0.8);
-webkit-transform:rotateX(-90deg) translateZ(100px) rotate(180deg);
-moz-transform:rotateX(-90deg) translateZ(100px) rotate(180deg);
}
.box div.side6 {
background:rgba( 0, 171, 147, 0.8);
```

```
-webkit-transform:rotateX(90deg) translateZ(100px);
-moz-transform:rotateX(90deg) translateZ(100px);
}
@-moz-keyframes spin {
0% {-moz-transform: rotateX(0deg) rotateY(0deg);}
100% {-moz-transform: rotateX(1080deg) rotateY(360deg);}
}
@-webkit-keyframes spin {
0% {-webkit-transform: rotateX(0deg) rotateY(0deg);}
100% {-webkit-transform: rotateX(1080deg) rotateY(360deg);}
}
@-moz-keyframes rot {
0% {-moz-transform: rotate(0deg);}
100% {-moz-transform: rotate(360deg);}
}
@-webkit-keyframes rot {
0% {-webkit-transform: rotate(0deg);}
100% {-webkit-transform: rotate(360deg);}
}
</style>
<div id="holder">
<div class="box">
<div class="side1"></div>
<div class="side2"></div>
<div class="side3"></div>
<div class="side4"></div>
<div class="side5"></div>
<div class="side6"></div>
</div>
</div>
```

Задания

Задание 1. Изучите, как создать анимацию. Для этого сохраните код примера1 и примера2 как веб-страницы и отобразите с помощью браузера. Для правильного отображения примера2 необходимо иметь 6 графических файлов.

Задание 2. Разработайте 2–3 своих примера с анимацией (например, прыгающий шар, метеоритный дождь, выезжающая машинка и т.д.). Сохраните в новом файле.

Задание 3. Создайте на своем сайте (сайт резюме) выпадающее меню с названием «Портфолио» и создайте подпункты: лаб1, лаб2

Подпункты меню должны быть ссылками на созданные в процессе выполнения соответствующих работ ваши web-страницы.

Разместите сайт в облачном хранении и подкрепите в разделе ДО.СКИФ на выполненное задание.