

## 1.6. Лабораторная работа № 6. Анимация. Создание выпадающего меню при помощи HTML, CSS.

Цель: изучить свойства анимации, научиться создавать выпадающие меню при помощи HTML и CSS.

### Краткая теория

#### 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.

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.

#### Пример1:

```
<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:

```

<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>

```

## Создание горизонтального меню

Для понимания поочередности создания меню требуется разобраться в свойствах:

- `display: flex;`
- `transition;`
- `position.`

**Display** – многоцелевое свойство, определяет, как элемент должен быть показан в документе.

Свойство **display** содержит параметр **flex**, при помощи которого, списки элементов меню будут формироваться по горизонтали.

**Transition** – универсальное свойство, которое позволяет установить эффект перехода между двумя состояниями элемента, они могут быть определены с помощью псевдоэлемента **hover** или **active**.

**Position** – устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.

Виды позиционирования:

- `absolute` (указывает на абсолютное позиционирование);
- `fixed` (осуществляет привязку к указанной точке экрана);
- `relative` (положение элемента устанавливается относительно исходного места).

Важно отметить, что при позиционировании, дополнительно к выбранному виду, положение элемента задается свойствами **left**, **top**, **right** и **bottom**.

## Дополнительная теория к созданию меню

В настоящее время для разработки различных интернет-средств используются нововведения. Создание меню не является исключением. При его разработке для визуального усовершенствования блока в примере ниже будет использовано следующее:

- иконки
- графическая визуализация
- шрифты Google

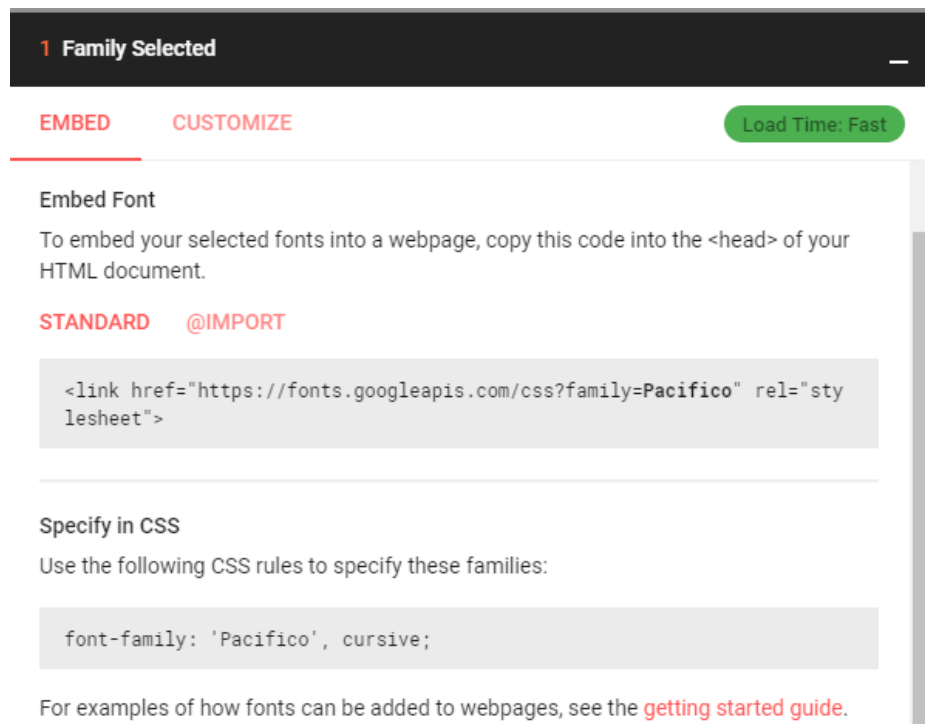
Для добавления иконок на сайт можно использовать некоторые интернет средства, при помощи которых можно получить код изображения, нужного

для тех или иных графических нюансов. Так же иконки можно создавать в графических редакторах: *CorelDraw*, *Adobe Illustrator*, *Adobe Photoshop* и другие. Для получения кода в этих программах следует сохранить созданное изображение в формате svg, далее открыть код и изъять блок `<svg></svg>` с его содержимым.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg enable-background="new 0 0 2602.652 1545.355" version="1.1" viewBox="0 0 2602.652 1545.355">
  <path d="m2118.8 521.66c189.39 34.204 347.78 89.099 462.49 173.14h-52.822c-119.05-58.32-260.27-119.05-419.32-119.05" />
</svg>
```

Например, можно использовать интернет средство для иконок: <http://fontawesome.io/icon/>.

Для получения онлайн-шрифта Google Fonts нужно найти непосредственно сам источник в интернете, выбрать нужный шрифт, подключить через онлайн библиотеку, а затем описать в стилях.



Графическая визуализация в данном коде, представляет собой саму модель построения блоков, а так же все элементы дизайна.

В данном коде имеется графическая визуализация блоков меню, в ней используются градиентные подборки, которые можно создать при помощи специальных **CSS-генераторов**, к примеру: <http://www.colorzilla.com/gradient-editor/>.

```
/* Permalink - use to edit and share this gradient: http://colorzilla.com/gradient-editor/
background: rgb(201,201,201); /* Old browsers */
background: -moz-linear-gradient(top, rgba(201,201,201,1) 0%, rgba(246,246,246,1) 2%,
background: -webkit-linear-gradient(top, rgba(201,201,201,1) 0%,rgba(246,246,246,1) 2%,
background: linear-gradient(to bottom, rgba(201,201,201,1) 0%,rgba(246,246,246,1) 2%,r
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#c9c9c9', endColorst
```

## Пример оформления:

```
<span class="menu">
<ul>
  <li><a href="#"><i class="fa fa-home"></i>Главная</a></li>
  <li><a href="#"><i class="fa fa-shopping-cart">
</i>Продукция</a>
  <ul>
    <li><a href="#">Одежда</a>
    <ul>
      <li><a href="#">Обувь</a></li>
      <li><a href="#">Куртки</a></li>
      <li><a href="#">Брюки</a></li>
    </ul>
    </li>
  </ul>
</li>
</ul>
</span>
```

## Задания к лабораторной работе № 6

**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 выпадающее меню. Подпункты меню должны быть ссылками на созданные в процессе выполнения соответствующих работ ваши web-страницы. Названия подпунктов придумайте сами.

Порядок выполнения:

- 1) Создайте html файл, который будет содержать элемент *span* с вложенным внутри *html-списком*, каждый элемент которого будет содержать пункты меню и субменю.
- 2) Создайте класс, который будет отбирать ссылки от *<ul>*, *<li>*, *<a>* и делать их блочными элементами при помощи *display: block*.
- 3) Примените к элементам *<ul>* свойство *display: flex*.
- 4) При наведении на элемент *<li>* должно быть использовано свойство *display: block*.

- 5) При наведении `<li>` и `<a>`, которые формируют полосу меню, должен изменяться цвет, а так же установите задержку при помощи свойства **transition** (*transition: all 0.3s ease;*).
- 6) Отредактируйте поля, руководствуясь информацией из раздела.