

Лабораторная работа № 5 «Создание графического меню»

Цель работы:

1. Изучение возможностей языка JavaScript по созданию в Web-странице графического меню.
2. Получение практических навыков по созданию графических меню с разными свойствами.

Оборудование и ПО: ПК, операционная система Windows7, офисный пакет Microsoft.

Порядок выполнения работы:

1. Ознакомление с методическими рекомендациями по созданию в Web-странице графических меню с разными свойствами. *Просмотр обучающих роликов к лабораторной работе № 5.*
2. Выполнение практической части задания в соответствии с рекомендациями разделов методического пособия.
3. Ответы на контрольные вопросы.
4. Оформление отчета.

I. Теоретическая часть и рекомендации

В данном методическом пособии будет рассмотрено три примера по типу создания графических меню:

- Вертикальное меню
- Горизонтальное меню
- Выпадающее меню

Если создаваемый Web-сайт не ограничивается одной веб-страницей, то стоит подумать о добавлении панели навигации (меню).

Меню - раздел веб-сайта, предназначенный помочь посетителю перемещаться по сайту. Любое меню представляет собой список ссылок, ведущих на внутренние страницы сайта.

Самым простым способом добавить панель навигации на сайт является создание меню с помощью CSS и HTML.

1.1. Вертикальное меню

Первым шагом создания вертикального меню будет создание маркированного списка. Также нам нужно будет иметь возможность идентифицировать список, поэтому мы добавим к нему атрибут id с идентификатором "navbar". Каждый элемент `` нашего списка будет содержать по одной ссылке:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a></li>
  <li><a href="#">О нас</a></li>
</ul>
```

Наша следующая задача заключается в сбросе стилей списка, установленных по умолчанию. Нам нужно убрать внешние и внутренние отступы у самого списка и маркеры у пунктов списка. Затем зададим нужную ширину:

```
#navbar {
  margin: 0;
  padding: 0;
  list-style-type: none;
  width: 100px;
}
```

Теперь пришло время стилизовать сами ссылки. Мы добавим к ним фоновый цвет, изменим параметры текста: цвет, размер и насыщенность шрифта, уберем подчеркивание, добавим небольшие отступы и переопределим отображение элемента `<a>` со строчного на блочный. Дополнительно были добавлены левая и нижняя рамки к пунктам списка.

Самой важной частью наших изменений является переопределение строчных элементов на блочные. Теперь наши ссылки занимают все доступное пространство пунктов списка, то есть для перехода по ссылке нам больше не нужно наводить курсор точно на текст.

```
#navbar a {
  background-color: #949494;
  color: #fff;
  padding: 5px;
  text-decoration: none;
  font-weight: bold;
  border-left: 5px solid #33ADFF;
  display: block;
}
#navbar li {
  border-left: 10px solid #666;
  border-bottom: 1px solid #666;
}
```

Мы объединили весь код, описанный выше, в один пример, теперь, нажав на кнопку попробовать, вы можете перейти на страницу с примером и увидеть результат:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Название документа</title>
  <style>
    #navbar {
      margin: 0;
      padding: 0;
      list-style-type: none;
      width: 100px;
    }
    #navbar li {
      border-left: 10px solid #666;
      border-bottom: 1px solid #666;
    }
    #navbar a {
      background-color: #949494;
      color: #fff;
      padding: 5px;
      text-decoration: none;
      font-weight: bold;
      border-left: 5px solid #33ADFF;
      display: block;
    }
  </style>
</head>
<body>
  <ul id="navbar">
    <li><a href="#">Главная</a></li>
    <li><a href="#">Новости</a></li>
    <li><a href="#">Контакты</a></li>
    <li><a href="#">О нас</a></li>
  </ul>
</body>
</html>
```

При наведении курсора мыши на пункт меню его внешний вид может изменяться, привлекая к себе внимание пользователя. Создать такой эффект можно с помощью псевдо-класса `:hover`.

Вернемся к рассмотренному ранее примеру вертикального меню и добавим в таблицу стилей следующее правило:

```
#navbar a:hover {
  background-color: #666;
  border-left: 5px solid #3333FF;
}
```

1.2. Горизонтальное меню

В предыдущем примере мы рассмотрели вертикальную панель навигации, которую чаще всего можно встретить на сайтах слева или справа от области с основным контентом. Однако меню с навигационными ссылками также часто располагается и по горизонтали в верхней части веб-страницы.

Горизонтальное меню можно создать путем стилизации обычного списка. Свойству `display` для элементов `` нужно присвоить значение `inline`, чтобы пункты списка располагались друг за другом.

Для размещения пунктов меню по горизонтали, сначала создадим маркированный список с ссылками:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a></li>
  <li><a href="#">О нас</a></li>
</ul>
```

Напишем для нашего списка пару правил, сбрасывающих стиль используемый для списков по умолчанию, и переопределим пункты списка с блочных на строчные:

```
#navbar {
  margin: 0;
  padding: 0;
  list-style-type: none;
}
#navbar li { display: inline; }
```

Теперь нам осталось лишь определить стилевое оформление для нашего горизонтального меню:

```
#navbar {
  margin: 0;
  padding: 0;
  list-style-type: none;
  border: 2px solid #0066FF;
  border-radius: 20px 5px;
  width: 550px;
  text-align: center;
  background-color: #33ADFF;
}
#navbar a {
  color: #fff;
  padding: 5px 10px;
  text-decoration: none;
  font-weight: bold;
  display: inline-block;
  width: 100px;
}
#navbar a:hover {
  border-radius: 20px 5px;
  background-color: #0066FF;
}
```

1.3. Выпадающее меню

Меню, которое мы будем создавать, будет иметь основные навигационные ссылки, расположенные в горизонтальной панели навигации, и подпункты, которые будут отображаться только после наведения курсора мыши на тот пункт меню, к которому эти подпункты относятся.

Сначала нам нужно создать HTML-структуру нашего меню. Основные навигационные ссылки мы поместим в маркированный список:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a></li>
  <li><a href="#">О нас</a></li>
</ul>
```

Подпункты мы разместим в отдельном списке, вложив его в элемент ``, который содержит родительскую ссылку относительно подпунктов. Теперь мы имеем четкую структуру нашей будущей панели навигации:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a>
    <ul>
      <li><a href="#">Адрес</a></li>
      <li><a href="#">Телефон</a></li>
      <li><a href="#">Email</a></li>
    </ul>
  </li>
  <li><a href="#">О нас</a></li>
</ul>
```

Теперь приступим к написанию CSS кода. Для начала необходимо скрыть список с подпунктами с помощью объявления `display: none;`, чтобы они не отображались на веб-странице все время. Для отображения подпунктов нам нужно чтобы при наведении на элемент `` список снова был преобразован в блочный элемент:

```
#navbar ul { display: none; }
#navbar li:hover ul { display: block; }
```

Убираем у обоих списков отступы и маркеры, установленные по умолчанию. Элементы списка с навигационными ссылками делаем плавающими, формируя горизонтальное меню, но для элементов списка, содержащих подпункты задаем `float: none;`, чтобы они отображались друг под другом.

```
#navbar, #navbar ul {
  margin: 0;
  padding: 0;
  list-style-type: none;
}
#navbar li { float: left; }
#navbar ul li { float: none; }
```

Затем нам нужно сделать так, чтобы наше выпадающее подменю не смещало контент, расположенный под панелью навигации, вниз. Для этого мы зададим пунктам списка `position: relative;`, а списку, содержащему подпункты `position: absolute;` и добавим свойство `top: 100%;`, чтобы абсолютно позиционированное подменю отображалось точно под ссылкой.

```
#navbar ul {
  display: none;
  position: absolute;
  top: 100%;
}
#navbar li {
  float: left;
```

```
    position: relative;
  }
  #navbar { height: 30px; }
```

Высота для родительского списка была добавлена специально, так как браузеры не учитывают в качестве содержимого элемента плавающий контент, то без добавления высоты наш список будет проигнорирован браузером и контент, следующий за списком, будет обтекать наше меню.

Теперь нам осталось стилизовать оба наших списка и выпадающее меню будет готово:

```
#navbar ul {
  display: none;
  background-color: #f90;
  position: absolute;
  top: 100%;
}
#navbar li:hover ul { display: block; }
#navbar, #navbar ul {
  margin: 0;
  padding: 0;
  list-style-type: none;
}
#navbar {
  height: 30px;
  background-color: #666;
  padding-left: 25px;
  min-width: 470px;
}
#navbar li {
  float: left;
  position: relative;
  height: 100%;
}
#navbar li a {
  display: block;
  padding: 6px;
  width: 100px;
  color: #fff;
  text-decoration: none;
  text-align: center;
}
#navbar ul li { float: none; }
#navbar li:hover { background-color: #f90; }
#navbar ul li:hover { background-color: #666; }
```

2. Контрольные вопросы:

1. Каким образом используются возможности JavaScript для создания графических меню в Web-странице?
2. Перечислите возможные варианты реализации графических меню.
3. Какие функции используются при реализации варианта вертикального меню?
4. Какие функции используются при реализации варианта горизонтального меню?
5. Какие функции используются при реализации варианта выпадающего меню?

3. Содержание отчета:

1. Наименование и цель лабораторной работы.
2. Результаты выполненных действий практической части в соответствии с заданием.
3. Ответы на контрольные вопросы.