

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

В.П. Часовских

Разработка и поддержка кроссплатформенных приложений

02.03.03 - Математическое обеспечение и администрирование информационных систем,
направленность (профиль) - разработка и администрирование информационных систем

Лабораторная работа 5 веб-приложение

Екатеринбург 2023

Веб-приложение – это объединение приложения и браузера! То есть, веб-приложение выполняется в веб-браузере. Соответственно, значительная доля вычислений производится на клиентской машине, а сервер занят только обработкой данных.

Создание структуры кода

С использования HTML создадим начальный код приложения:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>XXXXXX</title>
<link rel="stylesheet" href="css/bootstrap.css">
<link rel="stylesheet" href="css/base.css">
<!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.
min.js"></script>
<script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.
js"></script>
<![endif]-->
</head>
<body>
<script src="js/jquery-1.11.3.js"></script>
<script src="js/bootstrap.js"></script>
<script src="js/main.js"></script>
</body>
</html>
```

Добавление навигации

Создадим навигационную панель, поместив ее в начало тега

```
<nav class="navbar navbar-default navbar-fixed-top">
<div class="container">
<div class="navbar-header">
<a class="navbar-brand" href="webapp.html">

</a>
<button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#nav-menu"
aria-expanded="false">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<!-- <a class="btn btn-primary navbar-btn pull-left"
href="#" role="button">Sign up</a> -->
</div>
<div id="nav-menu" class="collapse navbar-collapse">
<ul class="nav navbar-nav">
</ul>
</div>
</div>
</nav>
```

Это простая навигационная панель с возможностью свертывания.

В ней есть логотипа: ``.

CSS-код для настройки логотипа приведен ниже:

```
.navbar-brand img {
height: 100%
}
```

Далее, внутри тега списка `ul.nav.navbar-nav` нужно создать элементы:

```
<ul class="nav navbar-nav">
<li>
<a href="#">
Home
</a>
</li>
```

```
<li>
```

```
<a href="#">
```

```
Notifications
```

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">
```

```
Messages
```

```
</a>
```

```
</li>
```

```
</ul>
```

Добавим значки для каждого пункта меню. Используем для этого Bootstrap Glyphicons. Добавление значков выделено жирным:

```
<ul class="nav navbar-nav">
```

```
<li>
```

```
<a href="#">
```

```
<span class="glyphicon glyphicon-home" aria-hidden=  
"true"></span>
```

```
Home
```

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">
```

```
<span class="glyphicon glyphicon-bell" aria-hidden=  
"true"></span>
```

```
Notifications
```

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">
```

```
<span class="glyphicon glyphicon-envelope" aria-hidden="true"></span>
```

Messages

```
</a>
```

```
</li>
```

```
</ul>
```

Посмотрите результат.

Добавление поля ввода для поиска

Добавим в навигационную панель поле ввода для поиска, применив два приема. Во-первых, поле ввода должно быть реализовано как группа элементов, чтобы присоединить значок лупы справа.

Во вторых, поле ввода должно быть прижато к правому, а не к левому краю панели `<nav>`. Создадим для этого форму, поместив ее после тега `ul.nav.navbar-nav`:

```
<div id="nav-menu" class="collapse navbar-collapse">
```

```
<ul class="nav navbar-nav">
```

```
...
```

```
</ul>
```

```
<form id="search" role="search">
```

```
<div class="input-group">
```

```
<input type="text" class="form-control"
```

```
  placeholder="Search...">
```

```
<span class="glyphicon glyphicon-search"
```

```
  aria-hidden="true"></span>
```

```
</div>
```

```
</form>
```

```
</div>
```

Добавим правила CSS, выравнивающие форму по правому краю с небольшим отступом:

```
nav form#search {
```

```

float: right;
padding: 0.5em;
}
nav form#search .glyphicon-search {
z-index: 99;
position: absolute;
right: 0.7em;
top: 50%;
margin-top: -0.44em;
}
nav form#search .input-group .form-control {
border-radius: 0.25em;
}

```

Теперь обновите веб-страницу и посмотрите результат.

Создание пунктов меню с миниатюрами

Добавим миниатюру, объединенную с кнопкой раскрывающегося меню непосредственно перед

form#search:

```

<div id="nav-options" class="btn-group pull-right">
<button type="button"
class="btn btn-default dropdown-toggle thumbnail"
data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">

</button>
<ul class="dropdown-menu">
<li><a href="#">Profile</a></li>
<li><a href="#">Setting</a></li>

```

```
<li role="separator" class="divider"></li>
<li><a href="#">Logout</a></li>
</ul>
</div>
```

Здесь использован шаблон кнопки с раскрывающимся списком. Вместо текста использовано изображение из учетной записи пользователя, а в тег `.btn-group` добавлен вспомогательный класс `.pull-right`. Так как кнопка определяется перед формой, на странице она появится после формы.

А теперь займемся правилами CSS: нужно изменить размер изображения и настроить поля и отступы:

```
#nav-options {
margin: 0.5em;
}
#nav-options button.thumbnail {
margin: 0
padding: 0;
}
#nav-options img {
max-height: 2.3em;
border-radius: 0.3em;
}
```

Добавление кнопки Tweet

Последним элементом навигационной панели станет кнопка Tweet.

Чтобы добавить ее, вставим следующий HTML-код непосредственно перед только что добавленной кнопкой:

```
<button id="tweet" class="btn btn-default pull-right">
<span class="glyphicon glyphicon-pencil" aria-hidden=
"true"></span> Tweet</button>
```

и добавим небольшие поля в CSS:

```
#tweet {  
margin: 0.5em;  
}
```

Посмотреть результат.

Настройка навигационной панели

Теперь, когда навигационная панель практически закончена, нужно настроить тему Bootstrap.

Настройка темы

Чтобы панель хоть немного отличалась от стандартной, окрасим ее фон в синий цвет. Для этого добавим несколько простых правил CSS:

```
.navbar-default {  
background-color: #2F92CA;  
}  
.navbar-default .navbar-nav > li > a {  
color: #FFF;  
}
```

После этого сделаем активным один из пунктов навигационного списка, добавив класс `.active` в первый элемент списка `nav` (пункт Home). Соответствующая строка выделена жирным:

```
<ul class="nav navbar-nav">  
<li class="active">  
<a href="#">  
<span class="glyphicon glyphicon-home" aria-hidden=  
"true"></span>  
Home  
</a>  
</li>
```



```
... <!-- другие теги li и остальной код -->
</ul>
```

И затем определим еще несколько правил CSS:

```
.navbar-default {
background-color: #2F92CA;
}
.navbar-default .navbar-nav > li > a {
color: #FFF;
}
.navbar-default .navbar-nav > .active > a {
background-color: transparent;
color: #FFF;
padding-bottom: 10px;
border-bottom: 5px solid #FFF;
}
```

Пункт Home будет находиться в активном состоянии. Чтобы обозначить это, в его изображение добавлена нижняя граница.

Исправление списка навигационной панели с помощью псевдо-классов

При наведении указателя мыши на любой элемент в списке навигации, он окрашивается неправильным цветом. Чтобы исправить это, воспользуемся переходами CSS3! Ниже приводится полный набор необходимых правил CSS:

```
.navbar-default {
background-color: #2F92CA;
}
.navbar-default .navbar-nav > li > a,
.navbar-default .navbar-nav > li > a:hover {
color: #FFF;
```

```

-webkit-transition: all 150ms ease-in-out;
-moz-transition: all 150ms ease-in-out;
-ms-transition: all 150ms ease-in-out;
-o-transition: all 150ms ease-in-out;
transition: all 150ms ease-in-out;
}
.navbar-default .navbar-nav > .active > a {
background-color: transparent;
color: #FFF;
padding-bottom: 0.62em;
border-bottom: 0.45em solid #FFF;}
.navbar-default .navbar-nav > .active > a:hover,
.navbar-default .navbar-nav > li > a:hover {
background-color: transparent;
color: #F3F3F3;
padding-bottom: 0.62em;
border-bottom: 0.45em solid #F3F3F3;

```

Добавленные в CSS3 переходы позволяют плавно изменять значения свойств. Чтобы определить переход, нужно указать свойство (в данном случае all), протяженность во времени и функцию (здесь используется ease-in-out)

С помощью этих правил мы изменили цвет по умолчанию навигационного списка, а определив переходы, создали красивый анимационный эффект – плавное появление нижней границы под элементом списка в момент наведения указателя мыши.

Решение проблем навигационной панели

Сейчас навигационная панель имеет три недостатка.

Это положение кнопки Tweet при отображении страницы в малых областях просмотра, сворачивание навигационного меню и цвет кнопки с пиктограммой бутербода.

Исправим кнопку Tweet! Для этого создадим еще один элемент, поместив его слева от кнопки сворачивания, и будем выводить его только при малом разрешении. Прежде всего, добавим класс `.hidden-xs` в кнопку Tweet:

```
<button id="tweet" class="btn btn-default pull-right hidden-xs">  
<span class="glyphicon glyphicon-pencil" aria-hidden=  
"true"></span>  
Tweet  
</button>
```

Затем, в `.navbar-header`, после `button.navbar-toggle`, добавим еще одну кнопку (выделена жирным):

```
<div class="navbar-header">  
<a class="navbar-brand" href="webapp.html">  
  
</a>  
<button type="button" class="navbar-toggle collapsed"  
data-toggle="collapse" data-target="#nav-menu"  
aria-expanded="false">  
<span class="sr-only">Toggle navigation</span>  
<span class="icon-bar"></span>  
<span class="icon-bar"></span>  
<span class="icon-bar"></span>  
</button>  
<button id="tweet" class="btn btn-default pull-right  
visible-xs-block">  
<span class="glyphicon glyphicon-pencil" aria-hidden=  
"true"></span>
```

Tweet

```
</button>
```

```
</div>
```

Итак, мы скрыли кнопку Tweet при отображении на сверхмалых устройствах и показали новую, в другом элементе. Выберите область просмотра для мобильных устройств и убедитесь, что положение кнопки исправлено.

Далее, исправим цвет кнопки с пиктограммой бутерброда, добавив следующие правила:

```
.navbar-header .navbar-toggle,  
.navbar-default .navbar-toggle:focus {  
background-color: #57A5D2;  
}  
.navbar-default .navbar-toggle:hover {  
background-color: #3986B3;  
}  
.navbar-default .navbar-toggle .icon-bar {  
background-color: #FFF;  
}
```

Наконец, настроим сворачивание навигационной панели с помощью вспомогательных классов Bootstrap.

Добавим класс `.hidden-xs` в элемент `.nav-options` и класс `.hidden-sm` в элемент `form#search`,

сделав их невидимыми на сверхмалых и малых устройствах, соответственно, как это было сделано выше с кнопкой Tweet:

```
<div id="nav-options" class="btn-group pull-right hidden-xs">
```

```
...
```

```
</div>
```

```
<form id="search" role="search" class="hidden-sm">
```

...

```
</form>
```

Затем, в навигационном списке `ul.nav.navbar-nav` создадим два пункта, которые заменят невидимые в текущей области просмотра:

```
<ul class="nav navbar-nav">
```

...

```
<!-- прочие скрытые элементы списка -->
```

```
<li class="visible-xs-inline">
```

```
<a href="#">
```

```
<span class="glyphicon glyphicon-user" aria-hidden="true"></span>
```

Profile

```
</a>
```

```
</li>
```

```
<li class="visible-xs-inline">
```

```
<a href="#">
```

```
<span class="glyphicon glyphicon-off" aria-hidden="true"></span>
```

Logout

```
</a>
```

```
</li>
```

```
</ul>
```

Здесь с помощью класса `.visible-xs-inline` создаются элементы, выстроенные в линию, которые видны при сверхмалых разрешениях.

В заключение уберем нижнюю границу активного пункта списка, так как она плохо смотрится при такой компоновке, и добавим левую, с помощью следующего медиа-запроса:

```
@media(max-width:34em){  
.navbar-default .navbar-nav > .active > a {  
border-bottom: none;
```

```
border-left: 0.45em solid #FFF;  
padding-left: 0.5em;  
}  
}
```