

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

СОЗДАНИЕ WEB-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ jQuery Mobile

Методические указания
к выполнению лабораторной работы
по дисциплине «Web-приложения
и мобильные информационные технологии»
для студентов очной формы обучения направления
09.03.01 «Информатика и вычислительная техника»

Севастополь
СевГУ
2020

УДК 004.428.2
С58

Р е ц е н з е н т:

Ю.Г. Шаталова – доцент кафедры «Фундаментальные дисциплины»
филиала ФГБОУ ВО «Государственный морской университет им. Ф.Ф. Ушакова»
в г. Севастополе, кандидат технических наук

Составитель: Лелеков С.Г.

С58 Создание Web-приложений на платформе jQuery Mobile :
методические указания к выполнению лабораторной работы по
дисциплине «Web-приложения и мобильные информационные
технологии» для студентов очной формы обучения направления
09.03.01 «Информатика и вычислительная техника» / Сост.
С.Г. Лелеков ; Министерство науки и высшего образования РФ,
Севастопольский государственный университет. – Севастополь:
СевГУ, 2020. – 23 с. – Текст : электронный.

Целью методических указаний является оказание помощи студентам в выполнении лабораторной работы по дисциплине «Web – приложения и мобильные информационные технологии».

Приведены варианты заданий, рекомендации по выполнению, требования к оформлению работы.

УДК 004.428.2

Методические указания рассмотрены и утверждены на заседании кафедры информационных технологий и компьютерных систем, протокол № 5 от 30 января 2020 г.

Рекомендовано Ученым советом Института информационных технологий и управления в технических системах, протокол № 4 от 3 марта 2020 г.

СОДЕРЖАНИЕ

Введение.....	4
1. Теоретические сведения	4
1.1. Создание веб-приложения jQuery Mobile	4
1.2. Структура контента.....	5
1.2.1. Нестандартные атрибуты data-*	5
1.2.2. Роли.....	5
1.2.3. Темы	5
1.2.4. Страница.....	5
1.2.5. Задание 1.	7
1.3. Архитектура web - приложения	8
1.3.1. Внутренние ссылки	8
1.3.2. Задание 2.	9
1.3.3. Абсолютные внешние ссылки.....	9
1.3.4. Мобильные специальные ссылки	10
1.3.5. Задание 3	11
1.4. Панели инструментов	11
1.4.1. Способы размещения панелей инструментов	11
1.4.2. Задание 4	12
1.4.3. Добавление содержимого в заголовок	12
1.4.4. Добавление содержимого в нижний колонтитул	12
1.4.5. Панели навигации	13
1.4.6. Задание 5	14
1.5. Экономия рабочего пространства экрана	14
1.5.1. Сворачиваемое содержимое	14
1.5.2. Модель «Аккордеон»	14
1.5.3. Сетки компоновки	16
1.5.4. Задание 6.	17
1.6. Кнопки.....	17
1.6.1. Способы создания кнопок.	17
1.6.2.Значки.....	18
2. Порядок выполнения работы	19
3. Список контрольных вопросов	20
Библиографический список.....	20
ПРИЛОЖЕНИЕ. Варианты заданий.....	21

Введение

Дисциплина «Web-приложения и мобильные информационные технологии» призвана содействовать ознакомлению студентов с компьютерными мобильными телекоммуникациями и возможными подходами к разработке гипертекстовых документов, предназначенных для публикации в глобальной компьютерной сети Internet, ориентированных на мобильные устройства.

Лабораторная работа «Создание web-приложений на платформе jQuery Mobile» позволяет студентам выработать и закрепить практические навыки Web-конструирования и Web-программирования для мобильных устройств.

1. Теоретические сведения

1.1. Создание веб-приложения jQuery Mobile

Для создания веб-приложения jQuery Mobile нужно создать пустой файл в формате HTML5.

Файл должен содержать код JavaScript и CSS-ресурсы (собственные или расположенные в сети доставки CDN) внутри тега <head> и тег <meta> для определения окна просмотра (листинг 1):

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Your Title</title>
    <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css"
/>
    <script src="http://code.jquery.com/jquery-
1.8.3.min.js"></script>
    <script
src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-
1.3.2.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-
scale=1">
</head>
<body>
<!-- Content -->
</body>
</html>
```

Листинг 1. Подключение фреймворка jQuery Mobile

Окно просмотра - это область, в которой находится страница. Можно указать ее ширину и высоту, причем она может быть больше или меньше видимой области экрана. В таком случае будут задействованы функции масштабирования в мобильном браузере. В оптимальном случае видимая область стартовой страницы равна по ширине экрану устройства. Тогда вы можете сообщить браузеру, что начальный масштаб должен быть 1:1 (отношение области окна просмотра к видимой области).

1.2. Структура контента

1.2.1. Нестандартные атрибуты data-*

Платформа jQuery Mobile использует нестандартные атрибуты data-*. Возможность применения нестандартных атрибутов в HTML-теге является особенностью HTML5 согласно спецификации W3C. Эта функциональная возможность HTML5 позволяет определять любые атрибуты, которые желательно добавить к тегу, без нарушения корректности HTML-документа.

Важная особенность нестандартных атрибутов состоит в том, что они работают в браузерах, не поддерживающих HTML5, и не вызывают никаких проблем.

1.2.2. Роли

На платформе jQuery Mobile используется стандартная HTML-разметка, в частности тег <div>. Для указания того, что платформа должна делать с тегом <div>, определяют его роль с помощью атрибута data-role. Например: <div data-role="page">

Основные роли, доступные в jQuery Mobile 1.0, перечислены в табл.1.

Таблица 1. Основные роли, доступные в jQuery Mobile

Роль	Описание
page	Определяет страницу, единицу показа содержимого в jQuery Mobile
header	Заголовок страницы
content	Содержимое страницы
footer	Нижний колонтитул страницы
navbar	Определяет строку навигации, как правило, внутри заголовка
button	Отображает визуальную кнопку
controlgroup	Отображает компонент
collapsible	Сворачиваемая панель с содержимым внутри страницы
collapsible-set	Группа сворачиваемых панелей (аккордеон)
fieldcontain	Контейнер для полей формы
listview	Содержимое нескольких элементов списка
dialog	Страница диалога
slider	Визуальный бегунок для булевых значений
nojs	Элемент, который будет скрыт в браузерах, совместимых с jQuery Mobile

1.2.3. Темы

Тема - это набор определений для компоновки, стилей и цветов. Каждая тема включает себя набор образцов цвета, который можно изменить в любом месте приложения. Образцы цвета создаются для того, чтобы можно было по-разному представлять элементы.

Образец цвета определяется для каждого HTML-элемента jQuery Mobile с помощью атрибута data-theme и буквенного обозначения, например: data-theme="e" обозначает цвет выделения.

Каждый элемент пользовательского интерфейса (страница, кнопка или компонент) может иметь свой образец цвета в рамках темы. Стандартные образцы тем обозначаются буквами от A до E с возможностью добавлять новые.

Образцы цвета подчиняются каскадному принципу. То есть, если в элементе контейнере определен образец цвета, этот образец применяется ко всем его потомкам, пока не будет явно определен другой образец.

1.2.4. Страница

Страница является основной единицей содержимого в jQuery Mobile.

Типичная страница разделяется на три части: заголовок (верхний колонтитул), содержимое и нижний колонтитул. Обязательной частью является только содержимое.

Каждая часть объявляется с помощью тега `<div>` с указанием его роли (листинг 2, рис. 1):

```
<div data-role="page">
  <div data-role="header">
  </div>
  <div data-role="content">
  </div>
  <div data-role="footer">
  </div>
</div>
```

Листинг 2. Шаблон страницы документа.

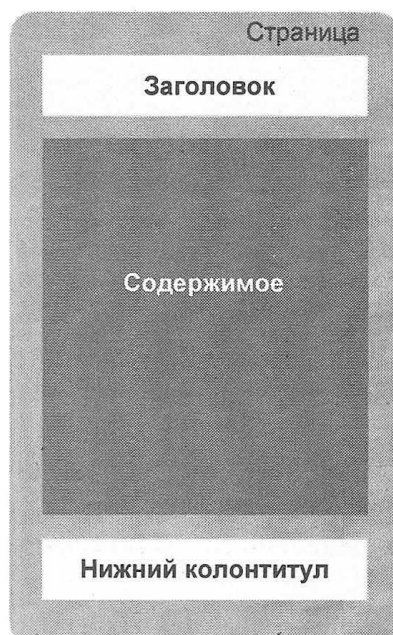


Рис. 1. Типичная страница документа jQuery Mobile

Любой элемент, т.е. страница, заголовок, нижний колонтитул и содержимое, имеет собственный образец цвета из текущей темы.

Платформа jQuery Mobile самостоятельно реагирует на смену ориентации (книжная или альбомная) на большинстве устройств, автоматически адаптируя пользовательский интерфейс к новому окну просмотра.

В заголовок и нижний колонтитул можно вставить любой HTML-код. Однако стандартная таблица стилей jQuery Mobile такова, что оптимальный результат отображения пользовательского интерфейса будет достигнут с помощью тега `<h1>` в заголовке и тега `<h4>` в нижнем колонтитуле.

Нижний колонтитул необязателен, а верхний обычно необходим для навигационных элементов интерфейса web-приложения.

Структура заголовка определена заранее, так что он состоит из левой части, собственно заголовка и правой части.

Левая и правая части обычно предназначены для кнопок.

Заголовок (центральная часть) будет автоматически взят из тега `<hX>`, например, из элемента `<h1>` или `<h2>`. Область для него ограничена, и он будет обрезан, если не

поместится. На большинстве устройств вместо обрезанной части будет многоточие (рис. 2).

То же самое справедливо и для нижнего колонтитула.

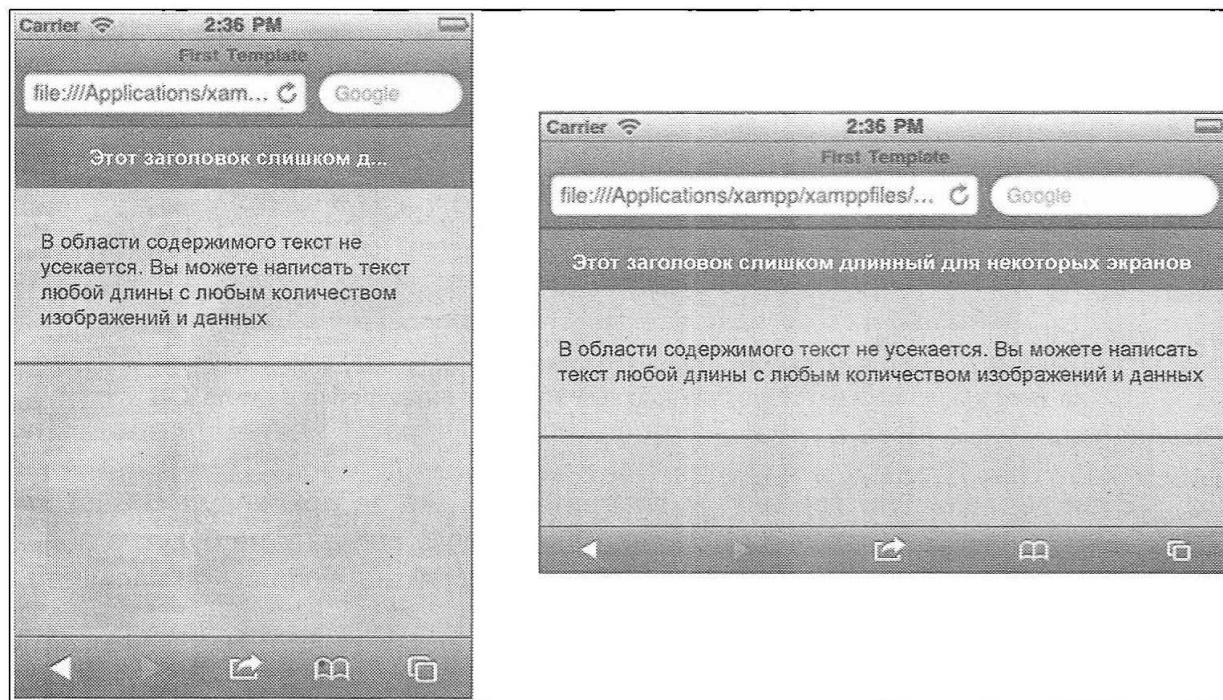


Рис. 2. Если заголовок не помещается, jQuery Mobile добавляет многоточие

В области содержимого может находиться любой HTML-код.

Изменим код из листинга 2 следующим образом (листинг 3):

```
<div data-role="page">
<div data-role="header">
    <h1>Our first webapp</h1>
</div>
<div data-role="content">
    <p>This is the main content of the page</p>
</div>
<div data-role="footer">
    <h4>More on mobilexweb.com</h4>
</div>
</div>
```

Листинг 3. Документ с использованием атрибутов jQuery Mobile

1.2.5. Задание 1.

1. Объедините листинг 1 с необходимыми файлами с примером страницы из листинга 3 в текстовом редакторе и сохраните в файле.
2. Замените содержимое верхнего колонтитула на свою фамилию и имя, контента – на любое приветствие, нижнего колонтитула – на шифр учебной группы.
3. Откройте файл в браузере.
4. Загрузите файл в свой смартфон и откройте его браузером смартфона (рис. 3).
5. Добавьте во все элементы страницы новые темы в соответствии с вариантами, приведенными в приложении.



Рис. 3. Скриншот результата просмотра документа задания 1

1.3. Архитектура web-приложения

Основной единицей платформы является страница. Один HTML-документ может содержать одну или несколько страниц в одном файле.

Можно переходить с одной страницы на другую в одном HTML-документе или на страницы во внешних HTML-документах, используя, например тег `<a>`.

Существует несколько типов гиперссылок:

- внутренние ссылки на другие страницы в одном (многостраничном) документе;
- внешние ссылки на страницы в других документах jQuery Mobile;
- абсолютные ссылки на документы, не имеющие отношения к jQuery Mobile;
- мобильные специальные ссылки.

1.3.1. Внутренние ссылки

Документ jQuery Mobile может включать в себя несколько страниц. Они должны быть оформлены как потомки элемента `<body>`. Чтобы иметь доступ к страницам, надо указать для каждой из них имя (атрибут `id`), пользуясь стандартным синтаксисом HTML.

Для создания ссылки на другую страницу в пределах одного документа достаточно воспользоваться конструкцией `#id` в атрибуте `href`, где `<id>` - имя целевой страницы. Например:

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

По умолчанию во всех переходах применяется переход "справа налево". Такой принцип хорош для навигации от общего к частному, когда пользователь хочет узнать подробности.

Платформа jQuery Mobile позволяет изменять или явно определять анимацию перехода для каждой смены страницы.

С этой целью используется атрибут `data-transition` внутри ссылки на новую страницу, например:

```
<a href="#page2" data-transition="pop">second page</a>
```

Доступны следующие переходы:

- `slide` - анимация "справа налево", выполняемая по умолчанию;
- `slideup` - анимация "снизу вверх", применяемая, в основном, для мобильных страниц;
- `slidedown` - анимация "сверху вниз";
- `pop` - новая страница появляется в виде точки в центре и разрастается на весь экран;

- fade - старая страница постепенно исчезает, в то время как "проявляется" новая;
- flip - двухмерное или трехмерное "переворачивание" страницы.
- turn - старая страница «поворачивается» к новой.
- flow – старая страница «отбрасывается».

Трехмерное переворачивание доступно только на некоторых устройствах, например, в системе iOS. На других устройствах, например, на работающих под управлением Android, визуально происходит двухмерное переворачивание.

Если вы хотите, чтобы анимированный переход воспроизводился в обратном направлении (при возврате на страницу), вы можете указать в соответствующей ссылке атрибут `data-direction="reverse"`. Платформа jQuery Mobile автоматически обеспечит воспроизведение анимации в обратном направлении.

Для возврата к предыдущей странице можно использовать как средства браузера, так и специальную кнопку «Назад».

Чтобы добавить программную кнопку Назад слева от заголовка, надо написать атрибут `data-add-back-btn="true"` на странице документа.

Конкретная надпись на кнопке определяется атрибутом `data-back-btn-text`, также на странице документа, а образец цвета можно задать атрибутом `data-back-btn-theme`. Например:

```
<div data-role="page" data-add-back-btn="true" data-back-btn-text="Назад" data-back-  
btn-theme="e">  
</div>
```

1.3.2. Задание 2.

1. Сформируйте двухстраничный документ.
2. На вторую страницу добавьте кнопку «Назад».
3. Воспользуйтесь эффектами анимации при межстраничных переходах в соответствии с вариантами, приведенными в приложении.
4. Опробуйте работу приложения в браузере мобильного устройства.

1.3.3. Абсолютные внешние ссылки

Иногда приходится делать ссылку на сайт или документ, не охваченный технологией jQuery Mobile. В этом случае необходимо явно определить абсолютную внешнюю ссылку, добавив атрибут `data-rel="external"` в тег `<a>`.

Например:

```
<a href="http://www.mobilexweb.com" data-rel="external">Check my Blog</a>
```

Существуют и другие ситуации, в которых ссылка интерпретируется как абсолютная внешняя. Например, это происходит при определении атрибута `target` или при ссылке на документ в другом домене, независимо от того, является ли он документом jQuery Mobile или нет:

```
<a href="http://www.mobilexweb.com" target="_blank">Check my Blog</a>  
<a href="http://www.otherdomain.com/whatever">Check my Blog</a>
```

Еще один способ сделать ссылку абсолютной внешней состоит в добавлении к ней атрибута `data-ajax="false"` (этот прием полезен для страниц, находящихся в том же домене):

```
<a href="otherpage.html" data-ajax="false">Other page</a>
```

1.3.4. Мобильные специальные ссылки

Мобильные специальные ссылки позволяют с помощью URI-схем совершать звонки и отправлять SMS-сообщения.

Телефонный звонок

```
<a href="tel:+79782299333">Позвони!</a>
```

Эта URI-схема была предложена в качестве стандарта RFC 5341 (<http://tools.ietf.org/html/rfc5341>)

Отправка сообщения по электронной почте

Почтовые приложения работают по классическому протоколу mailto.

Синтаксис обращения mailto:адрес?параметры. Список распознаваемых параметров может меняться от устройства к устройству, но он обычно включает в себя cc, bcc, subject и body. Параметры определяются в формате URL (ключ=значение&ключ=значение), причем значения должны быть кодированы (служебные знаки, кирилица).

Вот некоторые примеры:

```
<a href="mailto:info@rmobilexweb.com">Mail us</a>
```

```
<a href="mailto:info@rnbilexweb.com?subject=Contact%20from%20mobile">Mail us</a>
```

```
<a href="mailto:info@rmobilexweb.com?subject=Contact&body=This%20is%20the%20body">Mail us</a>
```

Надо отдавать себе отчет в том, что механизм mailto: не гарантирует отправку сообщения. Обычно он всего лишь открывает почтовое приложение, а пользователь должен подтвердить отправку, возможно, внося некоторые изменения.

Отправка SMS-сообщения

URI - схема имеет следующий синтаксис:

sms: // [<номер адресата>] [?параметры]

Как видите, номер адресата необязателен, так что вы можете открыть в устройстве окно для составления SMS-сообщения, не указав никаких параметров.

Параметры, участвующие в определении тела сообщения, совместимы не со всеми телефонами из соображений безопасности (чтобы у пользователя была возможность избежать отправки SMS-сообщений по завышенной цене). Как и в случае с сообщениями электронной почты, SMS-сообщение не отправляется автоматически после нажатия на ссылку. Ссылка лишь открывает окно приложения для составления SMS сообщений, и пользователь должен завершить процесс вручную.

Номер, на который отправляется сообщение, должен иметь международный формат, а если он является коротким, мы должны гарантировать, что абонент находится в соответствующей стране и пользуется услугами оператора, позволяющего звонить на этот номер.

Примеры:

```
<a href="sms:/// ">Send an SMS</a>
```

```
<!--Отправить SMS-сообщение -->
```

```
<a href="sms:///body=Visit%20the%20best%20site%20at%20http://mobilexweb.com">Invite a friend by SMS<a>
<!-- Пригласить друга по SMS -->
```

```
<a href="sms://+1111111111">Contact us by SMS</a>
<!-- Связитесь с нами по SMS -->
```

```
<a href="sms://+1111111111?body=Interested%20in%20Product%20AA2">More info
for product AA2</a>
<!-- Более подробная информация о товаре -->
```

1.3.5. Задание 3

1. Подготовьте страницу с возможностями связи с абонентом по телефону, электронной почте, SMS. В качестве адресатов выберите своих друзей.
2. Проверьте работу на мобильном устройстве.

1.4. Панели инструментов

1.4.1. Способы размещения панелей инструментов

Панели инструментов - это необязательные элементы web-приложения, определяющие верхние и нижние колонтитулы.

Каждая панель инструментов может быть размещена на страницы тремя способами:

- в режиме встраивания;
- в стандартном фиксированном режиме;
- в полноэкранном фиксированном режиме;

Режим встраивания устанавливается для каждой панели инструментов по умолчанию. Он означает, что заголовок и нижний колонтитул как бы встроены в страницу. Иначе говоря, если содержимое страницы не умещается на экране, нижний колонтитул будет скрыт и появится только после прокрутки, а заголовок будет виден, только когда на экране показано начало страницы.

Иногда бывает необходимо, чтобы панели инструментов были доступны все время, и для этой цели платформа jQuery Mobile предлагает полноэкранный и фиксированные режимы.

Для определения режима размещения панели инструментов предусмотрен атрибут data-position.

Фиксированная (определяемая значением fixed) панель инструментов jQuery Mobile будет размещена сверху (заголовок) или снизу (нижний колонтитул). Во время прокрутки содержимого страницы панель инструментов автоматически скрывается с эффектом постепенного исчезновения.

По окончании прокрутки фиксированная панель появляется в соответствующем месте, сверху или снизу.

Если пользователь прикоснется пальцем к неинтерактивной области страницы, фиксированные панели инструментов перейдут в режим встраивания, если до этого находились в фиксированном, и наоборот. Это позволяет пользователю переключать режимы прикосновением к экрану и освобождать место на нем, когда необходимо.

Для определения фиксированной панели инструментов мы пишем атрибут data-position="fixed".

Полноэкранная панель инструментов - это всего лишь скрытая фиксированная панель, которая появляется, когда пользователь касается экрана, и исчезает, когда пользователь касается экрана во второй раз. Содержимое в этом случае занимает весь экран, а если пользователю понадобится элемент панели инструментов, он просто прикоснется к экрану.

Для определения этого режима data-position="fixed" в коде панелей инструментов и data-fullscreen="true" в коде страницы (т. е. элемента, у которого атрибут data-role имеет значение page).

1.4.2. Задание 4

- Создайте html – документ достаточно большой по высоте из двух страниц.
- Опробуйте разные способы размещения панелей инструментов в соответствии с вариантом (см. Приложение).

1.4.3. Добавление содержимого в заголовок

Стандартный верхний колонтитул jQuery Mobile содержит собственно заголовок (элемент `<hX>`) и, возможно, одну или две кнопки, расположенные слева и/или справа.

Кнопка в заголовке - это всего лишь гиперссылка, определяемая элементом `<a>` внутри заголовка. Если мы напишем только один элемент `<a>`, он появится у левого края заголовка. Если мы создадим две кнопки, первая будет расположена у левого края, а вторая - у правого.

Для принудительного размещения кнопки справа или слева, можно использовать CSS-класс `class="ui-btn-right"` или, соответственно, `class="ui-btn-left"`.

В примере заголовка на листинге 4 мы увидим две кнопки, причем правая содержит значок:

```
<div data-role="header">
  <a href="logout">Log out</a>
  <h1>Title</h1>
  <a href="settings" data-icon="gear">Settings</a>
</div>
```

Листинг 4. Пример заголовка.

Как видно на рис. 4, мы можем использовать в кнопке любой из стандартных значков jQuery Mobile, добавив в код атрибут `data-icon`.

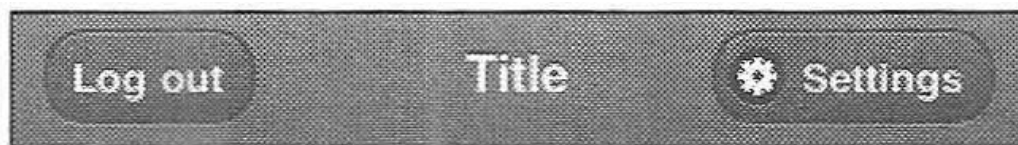


Рис. 4. Результат работы кода листинга 4

По умолчанию каждая кнопка панели инструментов наследует тему панели (а если эта тема не определена, то тему страницы). Чтобы кнопки отличались друг от друга, мы можем для одной из них задать атрибут `data-theme` и сменить ее цвет.

Добавление логотипа

Желание показать картинку или логотип вместо текста вполне обычно. Для этого лучшим способом будет применение тега `<img>` внутри тега `<h1>`:

```
<div data-role="header">
  <a href="cancel" data-icon="delete">Cancel</a>
  <h1></h1>
  <a href="save" data-icon="check" data-theme="b">Save</a>
</div>
```

Листинг 5. Добавление логотипа

Заголовок автоматически увеличится в соответствии с высотой изображения.

1.4.4. Добавление содержимого в нижний колонтитул

Нижний колонтитул является гораздо более гибким элементом, чем заголовок. Как и в заголовке, любой элемент `<a>` будет отображен в виде кнопки, хотя в нижнем

колонтитуле нет принудительного размещения кнопок слева или справа. Каждая кнопка встраивается в элемент и располагается после предыдущей. Можно разместить столько кнопок, сколько потребуется, как в обычной панели.

По умолчанию платформа не оставляет просвета между границей нижнего колонтитула и кнопками, поэтому для обеспечения привлекательного внешнего вида надо добавить в нижний колонтитул класс ui-bar.

На листинге 6 показан пример размещения четырех кнопок в нижнем колонтитуле.

```
<div data-role="footer" class="ui-bar">
    <a href="refresh">Refresh</a>
    <a href="filter">Filter</a>
    <a href="search">Search</a>
    <a href="add" data-theme="b">New Item</a>
</div>
```

Листинг 6. Размещение кнопок в нижнем колонтитуле.

1.4.5. Панели навигации

Панель навигации рассматривается как основной способ навигации пользователя по веб-приложению.

Панель навигации - это всего лишь контейнер (обычно элемент <div>), служащий оболочкой для неупорядоченного списка ссылок.

В качестве роли этого контейнера указывается значение navbar (см. листинг 7):

```
<div data-role="navbar">
<ul>
<li><a href="link1">Option 1</a>
<!-- ... -->
<li><a href="linkn">Option n</a>
</ul>
</div>
```

Листинг 7. Использование панели навигации.

Как видно на рис. 5, кнопки панели навигации внешне отличаются от стандартных кнопок, изображенных на рис. 4. Ширина кнопок рассчитывается в соответствии со следующим алгоритмом:

- одна кнопка на панели навигации занимает 100 % площади;
- две кнопки - по 50 %;
- три, четыре или пять кнопок - 33 %, 25 % или 20 % соответственно;
- шесть и более кнопок занимают несколько строчек по две кнопки в строке (по 50 % площади строки).

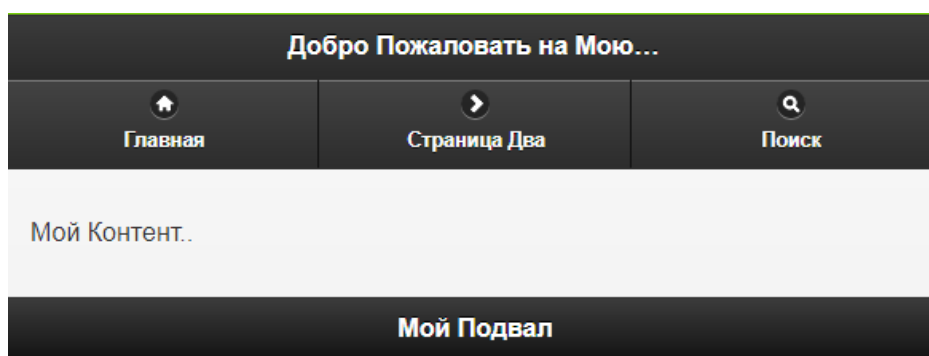


Рис. 5. Навигационная панель в верхнем колонтитуле

На рис. 5 также показана возможность использования пиктограмм. Для этого используется атрибут `data-icon` со стандартным именем или подключается собственный рисунок.

По умолчанию значок размещается над текстом. Если хотя бы у одного элемента навигации есть значок, высота всей панели навигации соответственно увеличивается.

На каждой панели навигации имеется выделенный элемент, в котором используется класс `ui-btn-active`. Активный элемент внешне контрастирует среди элементов пользовательского интерфейса, отображаемых согласно текущей теме (в теме по умолчанию он имеет голубой цвет, рис. 6).

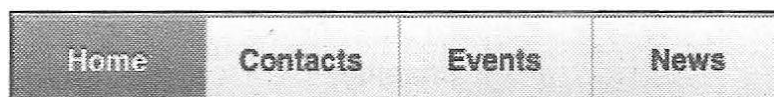


Рис. 6. Пример выделения элемента навигационной панели

1.4.6. Задание 5

Подготовить документ с использованием оформления верхнего и нижнего колонтитулов. Количество кнопок и их положение определяется вариантом (см. приложение).

1. Создайте собственный заголовок с использованием логотипа.
2. В нижнем колонтитуле использовать панель навигации.
3. Опробовать работу на мобильном устройстве.

1.5. Экономия рабочего пространства экрана

1.5.1. Сворачиваемое содержимое

Для создания сворачиваемого содержимого достаточно определить контейнер с атрибутом `data-role="collapsible"`. В этом контейнере должен находиться элемент `<hX>`, который будет всегда присутствовать как заголовок, играющий роль кнопки Скрыть/Показать.

Сворачиваемое содержимое - это любой HTML-код внутри контейнера, кроме упомянутого заголовка.

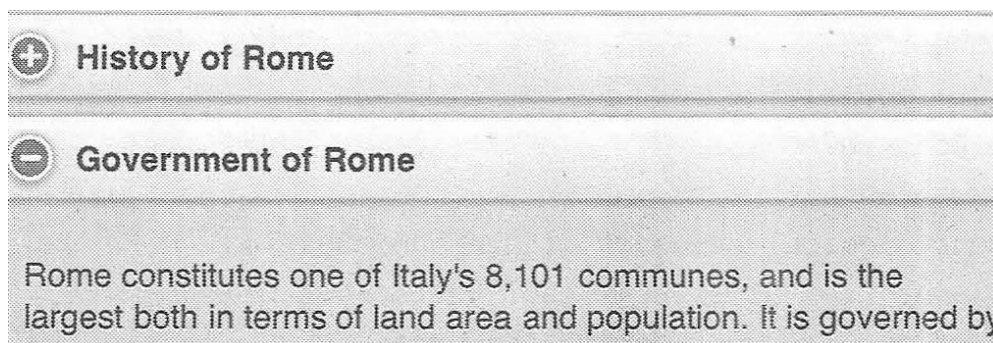


Рис. 7. Сворачиваемый контент

По умолчанию jQuery Mobile не показывает сворачиваемое содержимое при загрузке страницы. Мы можем изменить это умолчание с помощью атрибута `data-collapsed="false"` у контейнера.

1.5.2. Модель «Аккордеон»

Существует еще одна распространенная модель поведения насыщенного веб-приложения, связанная со сворачиванием содержимого: аккордеон. Она позволяет

группировать сворачиваемое содержимое так, чтобы была видна только одна панель, а остальные скрыты.

Платформа jQuery Mobile поддерживает такой тип компонентов в виде сворачиваемых наборов элементов. Набор представляет собой контейнер с атрибутом `data-role="collapsible-set"` и группу сворачиваемых панелей в качестве его потомков.

Шаблон сворачиваемого набора представлен на листинге 8:

```
<div data-role="collapsible-set">
  <div data-role="collapsible">
    <!-- Сворачиваемый заголовок и содержимое -->
  </div>
  <!-- Любой объем сворачиваемого содержимого -->
  <div data-role="collapsible">
    <!-- Сворачиваемый заголовок и содержимое -->
  </div>
</div>
```

Листинг 8. Шаблон пользовательского интерфейса «Аккордеон»

По умолчанию jQuery Mobile открывает последнюю сворачиваемую панель в сворачиваемом наборе. Если вы хотите, чтобы по умолчанию открывалась другая панель, поставьте атрибут `data-collapsed="false"` у соответствующего ей элемента и `data-collapsed="true"` у всех остальных элементов.

Пример использования метода «Аккордеон» представлен на листинге 9.

```
<div data-role="page" id="home">
  <div data-role="header">
    <h1>@firt</h1>
  </div>
  <div data-role="content" data-theme="b">
    <!-- Здесь определяется весь сворачиваемый набор (аккордеон) -
->
    <div data-role="collapsible-set">
      <div data-role="collapsible" data-collapsed="false">
        <h2>Books</h2>
        <ul>
          <li>Programming the Mobile Web</li>
          <li>jQuery Mobile: Up & Running</li>
          <li>Mobile HTML5</li>
        </ul>
      </div>
      <div data-role="collapsible" data-collapsed="true">
        <h2>Talks</h2>
        <ul>
          <li>Velocity Conference</li>
          <li>OSCON</li>
          <li>Mobile world Congress</li>
          <li>Google DevFest</li>
        </ul>
      </div>
    </div>
```

```

<!-- Конец сворачиваемого набора (аккордеон) -->
</div>
</div>

```

Листинг 9. Пример «Аккордеона».

Скриншоты отображения страницы представлены на рис. 8.

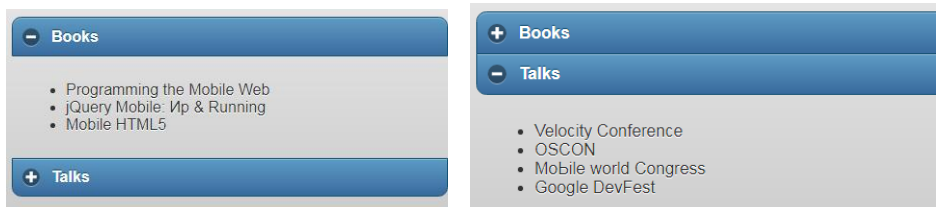


Рис. 8. Скриншоты отображения страницы из примера листинга 9

1.5.3. Сетки компоновки

Платформа jQuery Mobile предлагает ряд шаблонов для определения содержимого, выводимого в несколько столбцов. Эти шаблоны называются сетками компоновки.

Метод основан на применении классов CSS, определяющих области сетки и столбцы. Сетка может иметь от двух до 5 столбцов. Каждая сетка невидима, занимает 100 % ширины экрана, а в ее определении отсутствуют отступы и поля.

Чтобы создать сетку, достаточно использовать блочный контейнер, например `<div>`, с классом `ui-grid-a` для двух столбцов, `ui-grid-b` для трех, `ui-grid-c` для четырех и `ui-grid-d` для пяти столбцов. По умолчанию каждая сетка делит экран на столбцы одинаковой ширины.

Каждая ячейка сетки должна быть блочным контейнером с атрибутом `ui-block-буква`, где *буква* от *a* до *d* обозначает столбец сетки, с первого по пятый.

Рассмотрим пример компоновки из двух столбцов с использованием элемента `<section>`, появившегося в HTML5 (листинг 10):

```

<section class="ui-grid-a">
  <div class="ui-block-a">Column 1</div>
  <!-- Столбец 1 -->
  <div class="ui-block-b">Column 2</div>
  <!-- Столбец 1 -->
</section>
</div>

```

Листинг 10. Шаблон для создания сетки компоновки.

Сетки компоновки позволяют также реализовать табличное размещение. Это означает, что если мы добавим больше ячеек, чем имеется столбцов, то сможем эмулировать использование одной сетки в нескольких строках (листинг 11, рис. 9):

```

<section class="ui-grid-b">
  <div class="ui-block-a">Cell 1.1</div>
  <div class="ui-block-b">Cell 1.2</div>
  <div class="ui-block-c">Cell 1.3</div>
  <div class="ui-block-a">Cell 2.1</div>
  <div class="ui-block-b">Cell 2.2</div>
  <div class="ui-block-c">Cell 2.3</div>

```

</section>

Листинг 11. Создание табличного размещения.

Column 1		Column 2	
Cell 1.1	Cell 1.2	Cell 1.3	
Cell 2.1	Cell 2.2	Cell 2.3	

Рис. 9. Отображение контента с использованием столбцов

1.5.4. Задание 6

На основе листингов 9, 10, 11 создайте интерфейс «Аккордеон» и сетку компоновки. Варианты см. в приложении.

1.6. Кнопки

1.6.1. Способы создания кнопок

Для ссылок с одной страницы на другую или на внешнее содержимое можно применять любой элемент `<a>`. Элемент, как правило, является встроенным, а областью для щелчков служит текст. Это не вполне удобно пользователю сенсорного устройства. Именно поэтому платформа jQuery Mobile предоставляет нам кнопки.

Кнопка - это элемент пользовательского интерфейса, который представляет собой большую область для щелчков, содержащую текст и, возможно, значок.

Кнопку можно создать разными способами:

- с помощью элемента `<button>`;
- с помощью элемента `<input>`, который обычно отображается в виде кнопки и имеет атрибуты `type="button"`, `type="submit"`, `type="reset"` и `type="image"`;
- с помощью любого элемента `<a>` с атрибутом `data-role="button"`.

Кнопка jQuery Mobile обычно отображается с отцентрированной надписью, скругленными углами и тенью, в зависимости от того, насколько браузер совместим с форматом CSS3.

По умолчанию кнопка занимает всю ширину экрана, так что каждая кнопка выводится на отдельной строчке (см. рис. 10a)

```
<a href="#" data-role="button">Click me! </a>
<button data-theme="b">Click me too !</button>
<input type="button" value="Don' t forget about me !">
```

Можно создавать встроенные кнопки, которые не будут занимать всю ширину экрана. Для этого нужно применить к элементу атрибут `data-inline="true"`. (см. рис. 10b):

```
<a href="#" data-role="button" data-inline="true">Button 1</a >
<a href="#" data-role="button" data-inline="true">Button 2</a >
<a href="#" data-role="button" data-inline="true">Buttton 3</a>
```



Рис. 10а. Примеры создания кнопок

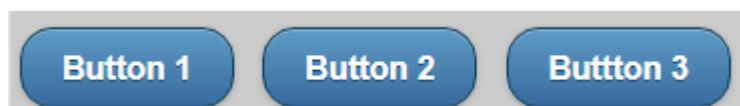


Рис. 10b. Примеры создания кнопок

Чтобы встроенные кнопки заняли всю ширину экрана, можно воспользоваться сетками компоновки и определить до пяти столбцов, в каждом из которых будет кнопка.

Если есть несколько кнопок, имеющих отношение друг к другу, их можно сгруппировать и получить другой пользовательский интерфейс, в котором каждая кнопка содержится в групповом контейнере. Для этого используется компонент, называемый группой элементов управления

Группа элементов управления - это просто контейнер, как правило, элемент `<div>` с атрибутом `data-role="controlgroup"`.

```
<div data-role="controlgroup">
  <a href="#" data-role="button">Button 1</a>
  <a href="#" data-role="button">Button 2</a>
  <a href="#" data-role="button">Button 3</a>
</div>
```

Группу можно расположить вертикально или горизонтально используя атрибут `data-type="horizontal"`.

1.6.2.Значки

Платформа jQuery Mobile имеет мощный механизм работы со значками, позволяющий применять темы к кнопкам и многим другим компонентам, на которых могут присутствовать пиктограммы.

В любой кнопке значок определяется атрибутом `data-icon`, в котором указано имя пиктограммы. Значки, доступные в jQuery Mobile 1.0, перечислены в табл. 2.

Таблица 2. Значки, доступные в jQuery Mobile

Описание значка	Значение атрибута «data-icon»
Стрелка влево	arrow-l
Стрелка вправо	arrow-r
Стрелка вверх	arrow-u
Стрелка вниз	arrow-d
Удалить (x)	delete
Плюс	plus
Минус	minus
Галочка	check
Шестеренка (настройки)	gear

Обновить	refresh
Вперед	forward
Назад	back
Сетка	grid
Звездочка	star
Внимание (пиктограмма предупреждения)	alert
Информация (i)	info
Главная страница	home
Поиск	search

На рис. 11 представлены все значки.

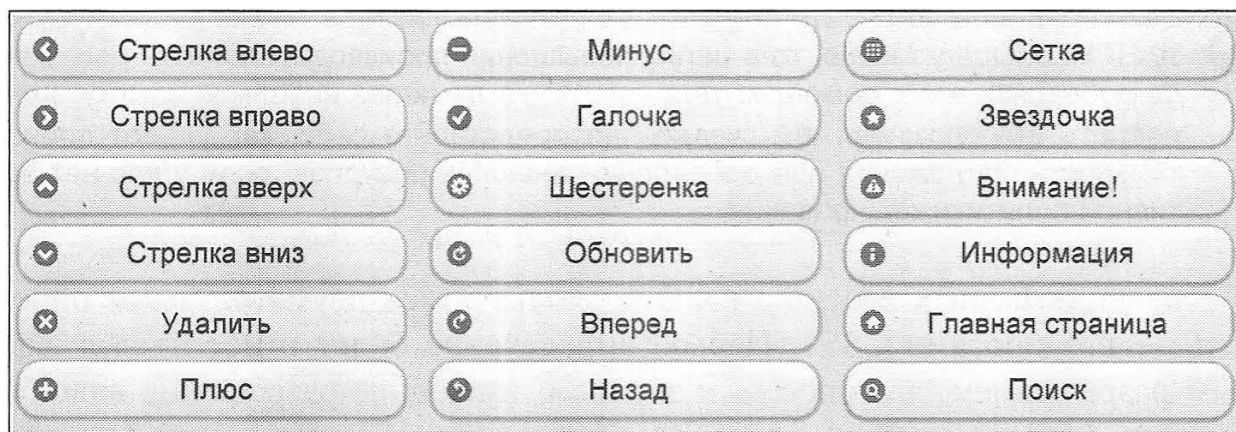


Рис. 11. Значки платформы jQuery Mobile

Можно использовать в приложении и нестандартный значок, для этого достаточно указать его уникальное имя в атрибуте `data-icon`. Рекомендуется придерживаться такой схемы наименования: `<имя-приложения>-<имя-значка>`, например, `myapp-tweet`.

Кроме того, платформа jQuery ожидает, что мы создадим CSS-класс с названием `ui-icon-<имя>`, например `ui-icon-myapp-tweet`, и укажем атрибут `background-image`. Чтобы избежать нежелательных последствий в пользовательском интерфейсе, необходимо нарисовать пиктограмму белым (или прозрачным) цветом и сохранить в виде изображения 18x18 пикселей в формате PNG-8 с альфа-прозрачностью. У значка не должно быть границы.

Код класса должен быть примерно таким:

```
<style>
.ui-icon-myapp-tweet { background-image: url(icons/tweet.png); }
</style>
```

2. Порядок выполнения работы

В ходе выполнения работы студент должен:

1. Ознакомиться с теоретическими сведениями.
2. Выполнить задания 1-6 в соответствии с вариантом (номер по списку учебной группы по модулю 15 плюс 1).
3. Подготовить отчет, включающий:
 - титульный лист;
 - задания;
 - тексты разработанных документов;
 - скриншоты результатов тестирования;
 - выводы.

3. Список контрольных вопросов

1. Назначение фреймворка jQuery Mobile.
2. Как используются нестандартные атрибуты data-*?
3. Сколько допускается использовать стандартных тем?
4. Какие существуют типы ссылок?
5. Какие существуют способы размещения панелей инструментов?
6. Что такое панели навигации?
7. Как можно сэкономить рабочее пространство экрана?
8. Какие существуют способы создания кнопок?

Библиографический список

1. Фиртман М. jQuery Mobile: разработка приложений для смартфонов и планшетов / Фиртман М. – Пер. с англ. – СПб.: БХВ-Петербург, 2013. – 256 с.: ил.
2. Вейл Э. HTML5. Разработка приложений для мобильных устройств / Э. Вейл. – СПб.: Питер, 2015. – 480 с.: ил.
3. Эспозито Д. Архитектура корпоративных мобильных решений / Д. Эспозито. – Пер. с англ. – М.: Изд-во «Русская редакция»; – СПб.: БХВ-Петербург, 2014. – 480 с.: ил.

ПРИЛОЖЕНИЕ

Варианты заданий

Задание 1

Темы оформления элементов страницы:

Вариант	Тема оформления		
	Верхний колонтитул	Содержимое	Нижний колонтитул
1	a	e	d
2	b	a	e
3	c	b	a
4	d	c	b
5	e	d	c
6	a	e	d
7	b	a	e
8	c	b	a
9	d	c	b
10	e	d	c
11	a	e	d
12	b	a	e
13	c	b	a
14	d	c	b
15	e	d	c

Задание 2

Анимированные переходы между страницами

Вариант	Переход	Реверс
1	slide	+
2	slideup	+
3	fade	+
4	flip	+
5	turn	+
6	flow	+
7	slide	-
8	slideup	-
9	fade	-
10	flip	-
11	turn	-
12	flow	-
13	flow	+
14	slide	-
15	slideup	+

Задание 4

Размещение панелей инструментов

Вариант	Номер страницы	Верхняя панель	Нижняя панель
1	1	Встроенная	Встроенная
2	1	Фиксированная	Фиксированная
3	1	Полноэкранная	Полноэкранная
4	1	Встроенная	Полноэкранная
5	1	Встроенная	Фиксированная
6	1	Фиксированная	Полноэкранная
7	1	Полноэкранная	Встроенная
8	1	Фиксированная	Встроенная
9	1	Полноэкранная	Фиксированная
10	2	Встроенная	Встроенная
11	2	Фиксированная	Фиксированная
12	2	Полноэкранная	Полноэкранная
13	2	Встроенная	Полноэкранная
14	2	Встроенная	Фиксированная
15	2	Фиксированная	Полноэкранная

Задание 5

Размещение кнопок в колонтитулах

Вариант	Верхний колонтитул		Нижний колонтитул	
	Количество кнопок	Положение кнопок	Количество кнопок	Активная кнопка
1	1	Слева	2	Первая
2	1	Справа	2	Вторая
3	2	Одна слева, другая справа	3	Первая
4	2	Обе слева	3	Вторая
5	2	Обе справа	3	Третья
6	1	Слева	4	Первая
7	1	Справа	4	Вторая
8	2	Одна слева, другая справа	4	Третья
9	2	Обе слева	4	Четвертая
10	2	Обе справа	2	Первая
11	1	Слева	3	Вторая
12	1	Справа	3	Первая
13	2	Одна слева, другая справа	4	Третья
14	2	Обе слева	4	Первая
15	2	Обе справа	4	Вторая

Задание 6

Варианты для самостоятельной работы

Вариант	Исходное состояние			Сетка компоновки	
	Компонент 1	Компонент 2	Компонент 3	Количество столбцов	Количество элементов
1	Развернут	Свернут	Свернут	5	5
2	Свернут	Развернут	Свернут	2	2
3	Свернут	Свернут	Развернут	3	3
4	Свернут	Свернут	Свернут	4	4
5	Развернут	Развернут	Свернут	5	10
6	Развернут	Свернут	Развернут	2	4
7	Развернут	Развернут	Свернут	3	6
8	Развернут	Развернут	Развернут	4	8
9	Развернут	Свернут	Свернут	5	15
10	Свернут	Развернут	Свернут	2	6
11	Свернут	Свернут	Развернут	3	9
12	Свернут	Свернут	Свернут	4	12
13	Развернут	Развернут	Свернут	5	20
14	Развернут	Свернут	Развернут	2	8
15	Развернут	Развернут	Свернут	3	12

СОЗДАНИЕ WEB-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ jQuery Mobile

*Методические указания
к выполнению лабораторной работы*

Составитель: **Лелеков** Сергей Георгиевич

В авторской редакции

Изд. № 56/2020. Объем 1,5 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»