**Виджеты jQuery UI**

jQuery UI предоставляет набор готовых виджетов предназначенных для создания пользовательского интерфейса веб-приложений.

Поведение виджетов может настраиваться с помощью передаваемых им опций.

Внешний вид виджетов может быть настроен:

1. С помощью выбора одной из стандартных тем доступных при скачивании библиотеки;
2. С помощью [themeroller'a](http://jqueryui.com/themeroller/);
3. Вручную путем редактирования файла jquery-ui-1.8.12.custom.css.

Общий вид создания виджетов jQuery UI выглядит примерно следующим образом:

1. Группировка элементов на странице особым образом (*индивидуально для каждого виджета*);
2. Применение к сгруппированным элементам специального метода, который превращает их в виджет.

**Виджет accordion**

**Виджет accordion** представляет собой группу объединенных выпадающих меню из которых только одно может быть отрыто одновременно.

Данный виджет используется для группировки информации на странице с целью экономии места.

**Синтаксис:**

/\* 1. Группируем элементы: \*/

<div id="accordion">

// 1.1. Определяем заголовок элемента

<h2><a href="#head1">Заголовок 1</a></h2>

/\* 1.2. Определяем содержимое элемента (значение атрибута id содержимого должно

совпадать со значением href заголовка) \*/

<div id="head1">Содержимое 1</div>

</div>

// 2. Применяем метод к сгруппированным элементам:

$("#accordion").accordion({опция1:значение1,опцияN:значениеN});

**Обратите внимание:** с помощью опций, передающихся методу, Вы можете настраивать дополнительные аспекты поведения виджетов. Виджетам может быть передано сразу несколько опций.

**Пример:**

$(document).ready(function() {

$("#accordion").accordion();

});

# Виджет autocomplete

**Виджет autocomplete** позволяет ускорить ввод данных в поле, отображая по мере введения определенные предположения. Выбрав одно из предположений пользователь может автоматически завершить ввод.

Предположения выводится в случае, если данные введенные пользователем совпадают со значением одного из элементов из списка данных.

Вы можете подключать к виджету как локальные (*т.е. определенные в скрипте на этой же странице*) так и удаленные списки (*т.е. находящиеся на удаленном сервере*).

Локальные списки удобны для хранения небольших наборов данных (*например список улиц города*), а удаленные списки подходят для хранения больших наборов данных (*например база данных всех городов мира*).

**Синтаксис:**

/\* 1. Определяем элемент input, в который будет производиться ввод информации \*/

<input id='auto' />

/\* 2. Превращаем input в виджет autocomplete \*/

$('#auto').autocomplete({опция1:значение1,опцияN:значениеN})

Подключить к виджету список данных можно с помощью опции **source**.

С помощью опции **minLength** Вы можете указать минимальное количество символов, которое должно содержать поле ввода прежде, чем поиск совпадений начнет выполнятся.

**Пример:**

$(document).ready(function() {

$("#auto1").autocomplete({source:["Дмитрий","Мария","Владимир","Алексей","Екатерина","Олег","Ольга"]});

});

# Виджет datepicker

Виджет **datepicker** привязывает к текстовому полю интерактивный календарь, который отображается когда поле становится активным.

Если пользователь щелкнет на какую-либо дату в календаре она будет автоматически введена в текстовое поле как будто он ввел ее вручную.

**Синтаксис:**

// 1. Создадим текстовое поле

<input type='text' id='datepicker' />

// 2. Привяжем к нему виджет datepicker

$('#datepicker').datepicker({опция1:значение1, опцияN:значениеN});

**Пример:**

$(document).ready(function() {

$("#datepicker").datepicker();

$("#datepicker1").datepicker({

monthNames:["Январь","Февраль","Март","Апрель","Май","Июнь","Июль","Август","Сентябрь","Октябрь","Ноябрь","Декабрь"],

dayNamesMin:["Вс","Пн","Вт","Ср","Чт","Пт","Сб"],

firstDay:1,

dateFormat:"dd.mm.yy"

});

});

**Оформление кнопок**

С помощью метода **button** Вы можете стилизовать:

* обычные кнопки (определяемые тэгами button и input type='button')
* кнопки отправления форм
* радио кнопки
* флажки
* ссылки

**Синтаксис:**

/\* Если Вы хотите оформить только один элемент, необходимо выбрать его с помощью

селектора и вызвать метод button: \*/

$("селектор").button({опция1:значение1, опцияN:значениеN});

/\* Если Вы оформляете группу элементов, необходимо вначале сгруппировать элементы

следующим образом: \*/

<div id="groupradio">

<input type="radio" name="radio" id="value1" />

<label for="value1">Радио кнопка 1</label>

<input type="radio" name="radio" id="value2" />

<label for="value2">Радио кнопка 2</label>

<input type="radio" name="radio" id="value3" />

<label for="value3">Радио кнопка 3</label>

</div>

// И затем вызвать метод buttonset:

$("#groupradio").buttonset({опция1:значение1, опцияN:значениеN});

**Пример:**

$(document).ready(function() {

$("#but1,#but2,#but3,#but4").button();

$("#group1,#group2").buttonset();

});

**Сделайте сами**

**Задание** выполните подпункты перечисленные ниже:

1. Исправьте ошибки в группировке элементов в задании для того, чтобы виджет accordion был отображен корректно.
2. Создайте на основе приведенных в задании данных виджет accordion.
3. Создайте виджет autocomplete, который будет производить поиск по списку приведенных в задании стран. Сделайте так, чтобы виджет начинал поиск только если длина ввода пользователя превышает 3 символа.
4. Оформите как кнопки элементы с id=el1, id=el2, id=el3.