Урок 6

jQuery UI и другие расширения jQuery

Изучение расширений для улучшения пользовательского интерфейса. Создание активных элементов при помощи jQuery UI.

jQuery UI

jQuery UI - это библиотека, основанная на библиотеке jQuery. Она реализует множество гибких плагинов, предоставляет структурированную и удобную систему оформления и инструменты для создания кастомизированных модулей с нуля, либо на основе существующих плагинов.

Перед тем, как начинать использовать все возможности библиотеки, jQuery UI, как и сам jQuery, сначала нужно подключить к странице применения.

Как и у jQuery, для jQuery UI есть два варианта подключения:

- локальное подключение – нужно скачать файл библиотеки с официального сайта;
- удаленное подключение - скачивание файла не требуется, нужно лишь подключить файл из репозитория Google.

При использовании первого способа на официальном сайте jQuery UI вы можете не только скачать стандартную комплектацию библиотеки, но и собрать свою собственную.

Стандартная комплектация jQuery UI компонуется из всех существующих плагинов с применённой темой оформления по умолчанию. Она скачивается по нажатию кнопки Download на сайте jQuery.

При необходимости получить кастомизированную комплектацию, нужно идти на сайт jQuery UI и следовать нижеперечисленным шагам:

1. **Выбор компонент.** Вероятно, не все компоненты библиотеки вам будут необходимы. Тогда убираем галочки с ненужных пунктов меню. При этом вы также сокращаете размер файла библиотеки, что ускоряет его загрузку. Таким образом меньший файл всегда предпочтителен.
2. **Выбор оформления.** В этом месте выбирается одна из предлагаемых тем оформления. В поле Theme вы вполне можете создать свою тему согласно необходимым цветам и стилям на целевом ресурсе.
3. **Выбор версии.** Также нужно выбрать версию jQuery UI. Разумеется, рекомендуется выбирать наиболее свежую.
4. **Скачать библиотеку.** После выполнения всех шагов нажать клавишу Download и скачать библиотеку для размещения на своём ресурсе.

Итак, независимо от того, скачали ли вы библиотеку в комплектации по умолчанию или собрали свою собственную, вам нужно подключить jQuery UI к странице сайта. Для этого нужно распаковать полученный архив и подключить файлы jquery-ui-версия.custom.css и jquery-ui-версия.custom.min.js в шапке (<head>) вашего сайта при помощи <script> и <link> соответственно.

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```

Интерактивность в jQuery UI

Библиотека предоставляет широкий спектр элементов, которые умеют взаимодействовать с пользователем. Например, можно создавать элементы Drag&Drop (их можно перемещать мышкой), Resizable (их размер можно менять), Sortable (их можно упорядочивать) и многие другие.

Drag

При помощи метода draggable вы сможете добавить выбранному элементу возможность перетаскивания.

```
$("my-selector").draggable({option1: value1});
```

Опции, указываемые в качестве аргументов, задают правила поведения элемента.

При создании такому элементу присваивается класс ui-draggable. В момент перетаскивания к нему добавляется класс ui-draggable-dragging.

Опции axis в параметрах даёт выбрать ось, по которой элемент может перетаскиваться. Это x (горизонтальная ось) и y (вертикальная ось).

```
$(document).ready(function() {
  $("my-selector1").draggable({axis:"x"});
  $("my-selector2").draggable({axis:"y"});
});
```

Опция cursor задаёт внешний вид указателя мыши в тот момент, когда ею перетаскивают draggable элемент.

```
$(document).ready(function() {
  $("my-selector1").draggable({cursor:"move"});
  $("my-selector2").draggable({cursor:"help"});
});
```

Опция helper помогает задавать вид элемента-помощника. Что это такое? Это особый элемент, который отображается в момент перетаскивания элемента. Без этой настройки по умолчанию будет отображаться сам перетаскиваемый блок.

```
$(document).ready(function() {
  $("my-selector1").draggable({helper:"clone"});
  $("my-selector2").draggable({helper:function(event){return $("<div>I'm a helper </div>")}});
});
```

Опция revert задаёт следующее поведение: возвращается ли элемент на исходную позицию по завершению процедуры перетаскивания.

```
$(document).ready(function() {
  $("my-selector1").draggable({revert:true});
});
```

Опция stack контролирует свойство z-index набора указанных в нём элементов. Как только перетаскиваемый элемент наводится на указанную группу, нужный набор просто спускается под перетаскиваемый блок.

Drop

Если есть drag, то должен быть и drop! В его создании помогает метод droppable. Он определяет область, которая готова принять перетаскиваемый элемент.

```
$("my-selector1").droppable({option1: value1});
```

Опции всё также определяют правила поведения элемента droppable, настраивая специальные аспекты поведения блока.

Опция-событие drop ждёт, когда перетаскиваемый элемент будет помещён в заданную область и сброшен там. Опция-событие over в свою очередь ждёт, когда перетаскиваемый элемент будет наведён на границы области.

```
$(document).ready(function(){
  $("my-selector-drag1").draggable({stack:"my-selector-drop1"});
  $("my-selector-drop1").droppable({
    over:function(){
      $("my-selector-drag1").css("background-color","#d7fa99");
      $("my-selector-drop1").css("background-color","#d7fa99");
    },
    drop:function(){
      $("my-selector-drag1").css("display","none");
      $("my-selector-drop1").css("background-color","#c4f66f");
      $("my-selector-drop1").html("Перемещение завершено успешно.");
      alert("Вы перетащили элемент с id=my-selector-drag1 в область с id=my-selector-drop1.");
    }
  });
});
```

Опция accept позволяет указать элементы, которые разрешены к принятию областью.

```
$(document).ready(function(){
  $("my-selector-drag1").draggable({stack:"my-selector-drop1"});
  $("my-selector-drag2").draggable({stack:"my-selector-drop1",revert:true});
  $("my-selector-drop1").droppable({
    accept:"#drag1",
    over:function(){
      $("my-selector-drag1").css("background-color","#d7fa99");
      $("my-selector-drop1").css("background-color","#d7fa99");
    },
    drop:function(){
      $("my-selector-drag1").css("display","none");
      $("my-selector-drop1").css("background-color","#c4f66f");
      $("my-selector-drop1").html("Перемещение завершено успешно.");
    }
  });
});
```

Resize

Следующий метод – это resizable. Он превращает элемент в растягиваемый блок.

```
$("my-selector1"). resizable({option1: value1});
```

Опции снова настраивают дополнительные аспекты поведения.

```
$(document).ready(function(){
  $("my-selector-resize1").resizable();
});
```

Опция animate – булева. Она может принимать значения true и false. При true размер выбранного элемента будет изменяться с плавной анимацией.

Опция helper, как и в drag, задаёт вид элемента-помощника, который показывает пользователю информацию о текущем размере элемента при растягивании.

```
$(document).ready(function(){
  $( my-selector-resize1").resizable({animate:true,helper:"ui-state-highlight"});
});
```

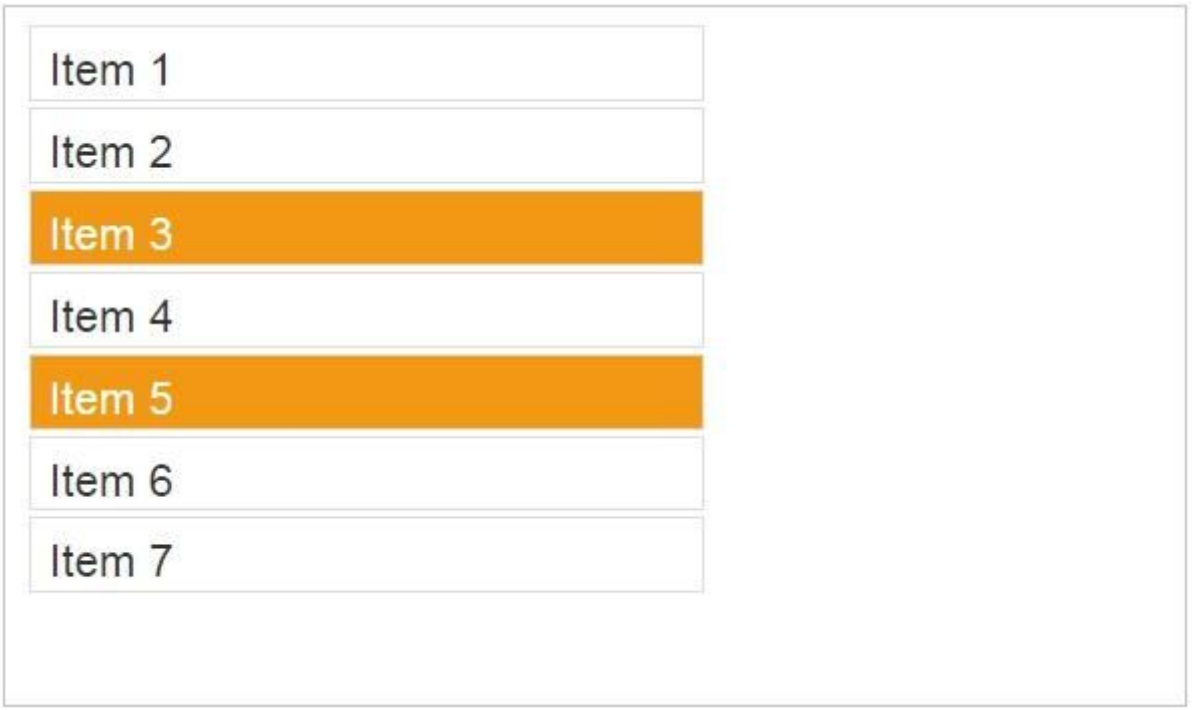
Выделяемые элементы

Метод selectable позволяет сделать из группы элементов в список для выбора значений.

```
$("my-selector").selectable({option1: value1});
```

Опции снова задают правила поведения.

Классы .ui-selecting и .ui-selected отвечают за CSS-правила для выбираемых и выбранных элементов соответственно.



Сортируемые элементы

Метод sortable делает элементы списка сортируемыми.

```
$("my-selector").sortable({option1: value1});
```

Позволяется создать два сортируемых списка. При этом элементы одного связанного списка можно перемещать в другой связанный список. Чтобы использовать эту возможность, нужно задать связываемым спискам одинаковый класс и указать его в опции connectWith.

```
$(document).ready(function(){
  $("my-selector-sort1, my-selector-sort2").sortable({connectWith:".connect"});
});
```

Виджеты jQuery UI

Помимо достаточно простых интерактивных действий jQuery UI предоставляет набор готовых удобных виджетов, которые могут создавать пользовательские интерфейсы веб-приложений.

Виджет accordion

Виджет accordion – это группа объединенных выпадающих элементов меню. При этом в один момент времени только один из них может быть открыт пользователем. Как правило, такой виджет применяется для компактного вывода графических меню с целью экономии места.

```
<div id="accordion">
<h2><a href="#head1">Заголовок 1</a></h2>
  <div id="head1">Содержимое 1</div>
<h2><a href="#head2">Заголовок 2</a></h2>
  <div id="head2">Содержимое 2</div>
</div>
```

```
$(document).ready(function() {
  $("#accordion").accordion({});
});
```

Виджет autocomplete

Одним из крайне полезных виджетов является виджет autocomplete. С ним вы не раз сталкивались при поиске информации в Google или Яндекс. По мере ввода данных в текстовое поле autocomplete

предлагает наиболее подходящие варианты, выбрав одно из которых можно быстро автоматически завершить ввод.

Важно помнить, что предположения о завершении поиска объекта выводятся только в случае, если данные, которые вводит пользователь, находят совпадение хотя бы в одном из списков данных. Обычно списки данных притягиваются в виджет с сервера, на котором хранится логика поиска. Сервер же возвращает подходящие списки, а виджет их стилизует и предлагает пользователю.

При этом локальные списки вполне пригодны для хранения небольших наборов данных (список тегов на сайте), а серверные списки гораздо лучше подходят для поиска по большим объемам данных (например, база данных городов мира).

```
<input id='my-autocomplete' />
```

```
$(document).ready(function() {
  $('#my-autocomplete').autocomplete({});
});
```

Списки данных подключаются к виджету через параметр source.

При указании опции minLength вы сможете задать минимальную длину строки, после которой стартует поиск совпадений. Поиск по коротким фразам будет генерировать очень большую нагрузку.

Виджет datepicker

Ещё один крайне удобный инструмент – это виджет datepicker, который попросту называют календарём. Это интерактивный блок для выбора даты и её автоматического ввода в указанное поле в нужном формате.

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

```
$(document).ready(function() {
$('#datepicker').datepicker({});
$("#datepicker").datepicker();
$("#datepicker").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

При помощи jQuery UI можно не только управлять полями ввода, но и задавать правила работы кнопкам. Это делается при помощи метода button. Он может управлять:

- обычными кнопками;
- кнопками отправки форм;
- радиокнопками;
- флажками;
- ссылками.

```
<div id="groupradio">
<input type="radio" name="radio" id="value1" />
<label for="value1">Радиокнопка 1</label>
<input type="radio" name="radio" id="value2" />
<label for="value2">Радиокнопка 2</label>
<input type="radio" name="radio" id="value3" />
<label for="value3">Радиокнопка 3</label>
</div>
```

```
$("селектор").button({});
$("#groupradio").buttonset({опция1:значение1, опцияN:значениеN});
$(document).ready(function() {
  $("#but1,#but2,#but3,#but4").button();
  $("#group1,#group2").buttonset();
});
```

Виджет tabs

Также часто бывает полезен вывод информации в виде вкладок, как, к примеру, листов в MS Excel. Виджет tabs решает эту задачу, группируя информацию на странице.

```
<div id="tabs">
<ul>
<li><a href="#tabs-1">Вкладка 1</a></li>
<li><a href="#tabs-2">Вкладка 2</a></li>
<li><a href="#tabs-3">Вкладка 3</a></li>
</ul>
<div id="tabs-1">
<p>Содержимое вкладки 1</p>
</div>
<div id="tabs-2">
<p>Содержимое вкладки 2.</p>
</div>
<div id="tabs-3">
<p>Содержимое вкладки 3.</p>
</div>
</div>
```

```
$("#tabs").tabs({});
$(document).ready(function() {
  $("#tabs").tabs();
});
```

Диалоговые окна

Вы все уже знакомы с функционалом alert и confirm. При помощи метода dialog можно создавать более насыщенные диалоговые окна.

По определению диалоговое окно всегда показывается поверх основных блоков страницы. Он состоит из заголовка и поля с содержимым.

Диалоговое окно способно к перетаскиванию, растяжению. Его можно закрывать кликом по иконке "x", как обычные окна в ОС. В случае, когда содержимое оказывается больше, чем размеры самого диалогового окна, то в окне появится полоса прокрутки.

```
<div id="dialog" title="Заголовок диалогового окна">
<p>Содержимое окна.</p>
</div>
```

```
$(document).ready(function() {
  $("#dialog2").dialog({width:400,height:300});
});
```

Опции width и height задают начальную высоту и ширину окна. Опция modal даёт сделать окно модальным (быть поверх контента и затемнять фон). Модальные окна обычно применяют для привлечения внимания со стороны пользователей. Это обычные рекламные «всплывашки», и работа с сайтом заблокирована, пока открыто модальное окно.

Опция autoOpen говорит, нужно ли открывать окно сразу же после создания.

Метод open открывает, а close – закрывает модальное диалоговое окно.

Опция buttons может добавлять различные кнопки прямо в окно, задавая их в виде {Название_Кнопки : function(){Код, который выполнится после нажатия на кнопку}}. В одно диалоговое окно, разумеется, может быть добавлено несколько кнопок.

```
$(document).ready(function() {
  $("#dialog1").dialog({autoOpen:false,buttons:{
    OK:function(){
      $(this).dialog("close");
      alert("Вы нажали OK");}}
  });
  $("#dialog2").dialog({autoOpen:false,buttons:{
    OK:function(){
      $(this).dialog("close");
      alert("Вы нажали OK");},
    Отмена:function(){
      $(this).dialog("close");
      alert("Вы нажали Отмена");}}
  });
  $("#but1").click(function(){
    $("#dialog1").dialog("open");
  });
  $("#but2").click(function(){
    $("#dialog2").dialog("open");
  });
});
```

Виджет slider

Виджета slider создаёт «ползунок». По сути – это обычное поле ввода, но данные в нём управляются графически. Примерно также, как вы управляете громкостью на своём компьютере.

```
<div id="slider"></div>
```

```
$(document).ready(function() {
  $("#slider").slider();
});
```

Эффекты jQuery UI

В jQuery вы можете добавлять элементам различные эффекты появления, исчезновения, анимации. Например, можно заставить некий блок постепенно исчезать.

jQuery UI также имеет такие возможности, но они значительно шире. В UI вы можете управлять пропаданием элементов со страницы пятнадцатью разными способами.

jQuery UI к тому же улучшает и расширяет функционал jQuery для управлением классами CSS.

Скрытие и отображение элементов с помощью эффектов jQuery UI

В jQuery UI эффекты используются со следующими методами:

- effect - применяет указанный эффект к выбранному элементу;
- hide - скрывает элемент с указанным эффектом;
- show - отображает элемент с указанным эффектом.

```
effect(effect,options,speed,callback);
hide(effect,options,speed,callback);
show(effect,options,speed,callback);
```

Параметр effect задаёт тип применяемого эффекта. Определённые эффекты (scale, transfer и size) для вызова дополнительно требуют задания опций options. Speed задаёт скорость эффекта в миллисекундах. Callback определяет функцию, которая выполняется после завершения применения эффекта.

Ниже перечислим доступные эффекты jQuery UI:

1. Blind Эффект "Ослепление".
2. Bounce Эффект "Отскок".
3. Clip Эффект "Отсечение".
4. Drop Эффект "Падение".
5. Explode Эффект "Взрыв".
6. Fade Эффект "Выцветание".
7. Fold Эффект "Складка".
8. Highlight Эффект "Освещение".

9. Puff Эффект "Рассеивание".
10. Pulsate Эффект "Пульсирование".
11. Scale Эффект "Масштабирование". С помощью опции percent:проценты вы можете задать на сколько процентов от текущего размера необходимо уменьшить или увеличить элемент.
12. Shake Эффект "Тряска".
13. Slide Эффект "Скольжение".
14. Size Эффект "Калибровка". С помощью опции to: и включенных в неё опций width:ширина_в_пикселях и height:высота_в_пикселях вы можете задать размеры, до которых необходимо "откалибровать" текущий элемент.
15. Transfer Эффект "Переход". С помощью опции to:id_или_class_элемента вы можете указать элемент, в который перейдёт текущий элемент. С помощью опции className:имя_класса вы можете оформить эффект перехода с помощью CSS.

jQuery UI анимация

Также jQuery UI улучшает возможности jQuery-анимации. Например, вы сможете создавать анимацию, манипулирующую цветом, применяя следующие CSS свойств:

- backgroundColor;
- borderColor;
- borderBottomColor;
- borderLeftColor;
- borderRightColor;
- borderTopColor;
- color;
- outlineColor.

```
$(document).ready(function() {
  $("#but1").click(function(){
    $("#testcontainer").animate({
      borderColor:"#EA3B53",
      borderWidth:3,
      backgroundColor:"#97D400",
      width:500,
      height:160},1500);
  });
});
```

Манипуляции с CSS-оформлением

Также в jQuery UI улучшена возможность управления CSS-классами. В JUI переключение можно делать постепенно и анимировано.

```
$(document).ready(function(){
  $("#but1").click(function(){
    $(".el1").addClass("newclass",1000);
  });
  $("#but2").click(function(){
    $(".el1").removeClass("newclass",1000);
  });
  $("#but3").click(function(){
    $(".el1").toggleClass("newclass",1000);
  });
  $("#but4").click(function(){
    $(".el1").switchClass("el1","newclass");
  });
});
```

Домашнее задание

1. На сайте в форме обратной связи добавьте следующие поля:
    a. поле даты рождения;
    b. ошибочные поля подсветить с помощью какого-нибудь эффекта, например, Bounce.
2. Все возвращаемые ошибки выводить с помощью виджета Dialog.
3. Создать карусель популярных товаров в шапке.
4. * С помощью jQuery UI добавить возможность перемещать товар прямо в корзину мышью.

Дополнительные материалы

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. http://api.jqueryui.com/