Стандартные анимации

- **.show() .hide()**
- Мгновенно показывает/скрывает выбранные элементы, установив их css-свойство display в none, не изменяя при этом их прозрачность и размеры.

- **.show(duration,[callback]) .hide(duration,[callback])**

duration — продолжительность выполнения анимации (появления или скрытия).

Может быть задана в миллисекундах (200 и 600 миллисекунд)

Может быть задана строковым значением 'fast' или 'slow'.

Если этот параметр не задан, анимация будет происходить мгновенно, элемент просто появится/исчезнет

callback — функция, заданная в качестве обработчика завершения анимации (появления или скрытия).

Стандартные анимации

- **.animate(properties, [duration], [easing], [callback])**
- Выполняет заданную пользователем анимацию, с выбранными элементами. Анимация происходит за счет плавного изменения CSS-свойств у элементов.

```
$('div').animate(
  {
    opacity: "hide",
    height: "hide"
  },
5000);
```

```
$("#right").click(function(){
  $(".block").animate({"left": "+=50px"}, "slow");
});
$("#left").click(function(){
  $(".block").animate({"left": "-=50px"}, "slow");
});
```

Стандартные анимации

- **.animate(properties, [duration], [easing], [callback])**
- Выполняет заданную пользователем анимацию, с выбранными элементами. Анимация происходит за счет плавного изменения CSS-свойств у элементов.

```
$("div").animate({left: "200px", top: "200px"}, 500);
```

- Данная анимация одновременно применяется к свойству left, для которого задано значение 200px, и к свойству top со значением 200px, продолжительность анимации задана 500ms.
- Для элемента можно задавать относительное перемещение при каждом вызове анимации с помощью операторов +=, -=, *=, /=, например,

```
$("div").animate({left: "+=200", top: "-=200"}, 500);
```

Стандартные анимации

- **. fadeIn(длительность, функция по завершении анимации)**
- Выполняет заданную пользователем анимацию, с выбранными элементами. Анимация происходит за счет плавного изменения CSS-свойств у элементов.

```
$("div").animate({left: "200px", top: "200px"}, 500);
```

- Данная анимация одновременно применяется к свойству left, для которого задано значение 200px, и к свойству top со значением 200px, продолжительность анимации задана 500ms.
- Для элемента можно задавать относительное перемещение при каждом вызове анимации с помощью операторов +=, -=, *=, /=, например,

```
$("div").animate({left: "+=200", top: "-=200"}, 500);
```

jQuery UI

•. **jQuery UI** представляет собой группу плагинов jQuery облегчающих создание интерфейса веб-приложений.

•**Подключение jQuery UI**

•Для того, чтобы воспользоваться возможностями плагинов jQuery UI их необходимо вначале подключить к странице, на которой они будут использоваться.

•Существуют два варианта подключения jQuery UI:

1. Локальное подключение. Данный способ требует скачивание специального файла с официального сайта;
2. Удаленное подключение. Данный способ не требует скачивание файла, а вместо этого использует его удаленно (данная услуга предоставляется компанией Google).

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;
3. Вручную путем редактирования файла jquery-ui-1.8.12.custom.css.

Общий вид создания виджетов jQuery UI выглядит примерно следующим образом:

Группировка элементов на странице особым образом (индивидуально для каждого виджета);

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.

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*).

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.

Оформление кнопок

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

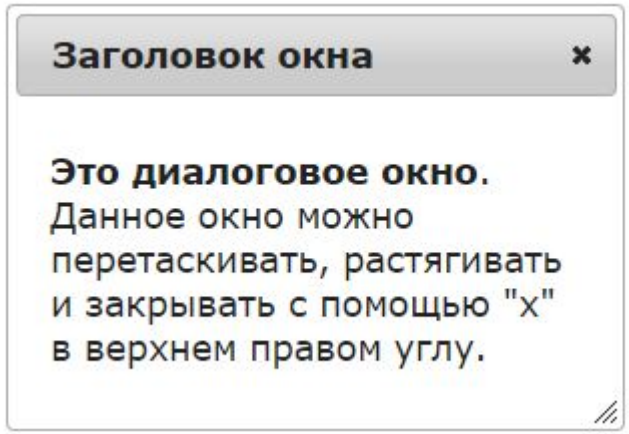
1. обычные кнопки (определяемые тэгами button и input type='button')
2. кнопки отправления форм
3. радио кнопки
4. флажки
5. ссылки

Виджет tabs

Виджет **tabs** как и виджет accordion используется для группировки информации на странице с целью экономии места.

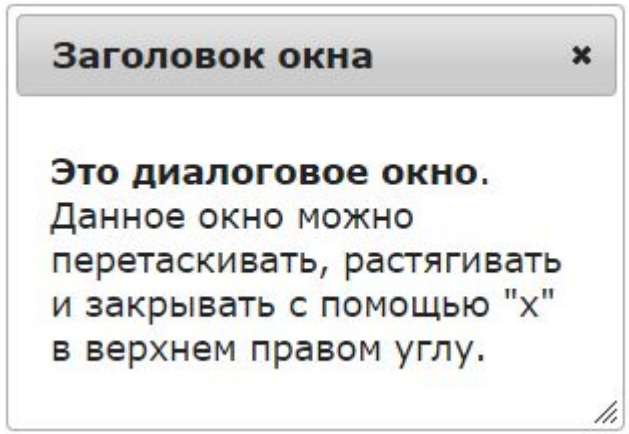
Диалоговые окна

- С помощью метода **dialog** Вы можете превратить выбранный элемент в **диалоговое окно**.
- Диалоговое окно отображается поверх текста страницы и содержит заголовок и поле с содержимым.
- Диалоговое окно можно перетаскивать, растягивать и закрывать с помощью иконки "x" в верхнем правом углу окна. Если содержимое превышает размеры диалогового окна, то полоса прокрутки будет автоматически отображена.



Диалоговые окна

- С помощью метода **dialog** Вы можете превратить выбранный элемент в **диалоговое окно**.
- Диалоговое окно отображается поверх текста страницы и содержит заголовок и поле с содержимым.
- Диалоговое окно можно перетаскивать, растягивать и закрывать с помощью иконки "x" в верхнем правом углу окна. Если содержимое превышает размеры диалогового окна, то полоса прокрутки будет автоматически отображена.



Методы создания AJAX запросов

- AJAX запросы - это асинхронные запросы к серверу позволяющие обновлять только ту часть страницы, которая содержит новую информацию, без необходимости обновлять страницу целиком.
- Использование AJAX запросов ускоряет загрузку страниц и снимает нагрузку с сервера.
- Все существующие в jQuery методы для создания AJAX запросов перечислены в таблице ниже:

Методы создания AJAX запросов

Метод	Описание
$.ajax()	Выполняет AJAX запрос.
ajaxComplete()	Определяет функцию, код которой будет выполнен когда AJAX запрос будет совершен.
ajaxError()	Определяет функцию, код которой будет выполнен если во время выполнения AJAX запроса произойдет ошибка.
ajaxSend()	Определяет функцию, код которой будет выполнен перед отправлением AJAX запроса на сервер.
$.ajaxSetup()	Позволяет установить данные для будущих AJAX запросов.
ajaxStart()	Определяет функцию, код которой будет выполнен перед тем, как первый AJAX запрос из группы запросов будет отправлен на сервер.
ajaxStop()	Определяет функцию, код которой будет выполнен, когда последний AJAX запрос из группы запросов будет совершен.

Методы создания AJAX запросов

Метод	Описание
ajaxSuccess()	Определяет функцию, код которой будет выполнен если AJAX запрос будет совершен успешно.
$.get()	Позволяет загрузить данные с сервера используя HTTP запрос GET.
$.getJSON()	Позволяет загрузить JSON - данные с сервера используя HTTP запрос GET.
$.getScript()	Позволяет загрузить с сервера JavaScript код и исполнить его.
load()	Позволяет загрузить данные с сервера и вставить их в содержимое выбранного HTML элемента.
$.param()	Позволяет создать сериализованное представление массива или объекта.
$.post()	Позволяет загрузить данные с сервера используя HTTP запрос POST.
serialize()	Позволяет закодировать группу элементов формы как строку для отправки с помощью AJAX запроса.
serializeArray()	Позволяет закодировать группу элементов формы как массив из их имен и значений.