JQUERY: EFFECTS AND FORMS

OVERVIEW

- Effects
- The .animate() method
- Form elements
- Form events and methods
- jQueryUI

JQUERY EFFECTS

- When you start using jQuery, the effects methods can enhance your web page with transitions and movement.
- Here you can see some of the jQuery effects that show or hide elements and their content.

Method	Effect
.show():	Displays selected elements
.hide():	Hides selected elements
.toggle():	Toggles between showing and hiding selected elements

JQUERY EFFECTS

- You can also animate elements, fade them in or out, or slide them up and down.

Method	Effect
.fadeIn():	Fades in selected elements making them opaque
.fadeOut():	Fades out selected elements making them transparent
.fadeTo():	Changes opacity of selected elements
.fadeToggle():	Hides or shows selected elements by changing their opacity (the opposite of their current state)
.slideUp():	Hides selected elements with a sliding motion
.slideDown():	Shows selected elements with a sliding motion
.slideToggle():	Hides or shows selected elements with a sliding motion (in the opposite direction to its current state)

JQUERY EFFECTS

- You can also create custom effects using the animate method.
- Stop and delay can be used to control the timing and execution of effects.

Method	Effect
.delay():	Delays execution of subsequent items in queue
.stop():	Stops an animation if it is currently running
.animate():	Creates custom animations (more on this later)

JQUERY EFFECTS EXAMPLE

- In this example it appears as if list items are faded into view when the page loads.
- Each item is faded out when it is clicked on.

JQUERY EFFECTS EXAMPLE

- In fact, the items are loaded normally along with the rest of the page, but then immediately hidden using jQuery.
- Once hidden, only then are they faded into view. This is so they will still be visible in browsers that do not have JavaScript enabled.
- In the first statement, the selector picks the <h2> element and hides it so that it can be animated in. The chosen effect to show the heading is the .slideDown() method.
- Note how the methods are chained; there is no need to make a new selection for each of the tasks.

```
$(function() {

  $('h2').hide().slideDown();

});
```

JQUERY EFFECTS EXAMPLE

- The second part causes the list of items to appear one by one.
  - Again, before they can be faded in, they must be hidden.
  - Then the .each() method is used to loop through each of the <li> elements in turn. You can see that this triggers an anonymous function.
  - Inside the anonymous function the index property acts as a counter indicating which <li> element is the current one.
  - The .delay() method creates a pause before the list item is shown. The delay is set, multiplying the index number by 700ms (otherwise all of the list items would appear at the same time).
  - Then it is faded in using the .fadeIn() method.

JQUERY EFFECTS EXAMPLE

```
$(function() {

  $('h2').hide().slideDown();

  $('li').hide().each(function(index) {
    $(this).delay(700 * index).fadeIn(700);
  });

});
```

JQUERY EFFECTS EXAMPLE

- The final part creates an event listener that waits for the user to click on a list item.
- When they do, it will fade that item out to remove it from the list (the fade will take 700 milliseconds).

```
$(function() {

  $('h2').hide().slideDown();

  $('li').hide().each(function(index) {
    $(this).delay(700 * index).fadeIn(700);
  });

  $('li').on('click', function() {
    $(this).fadeOut(700);
  });

});
```

THE ANIMATE METHOD

- The .animate() method allows you to create some of your own effects and animations by changing CSS properties.
- You can animate any CSS property whose value can be represented as a number, e.g. height, width and font-size (but not those whose value would be a string e.g. font-family or text-transform).
- The CSS properties are written using camelCase notation. For example, **border-left-top-radius** would become **borderLeftTopRadius**.

THE ANIMATE METHOD

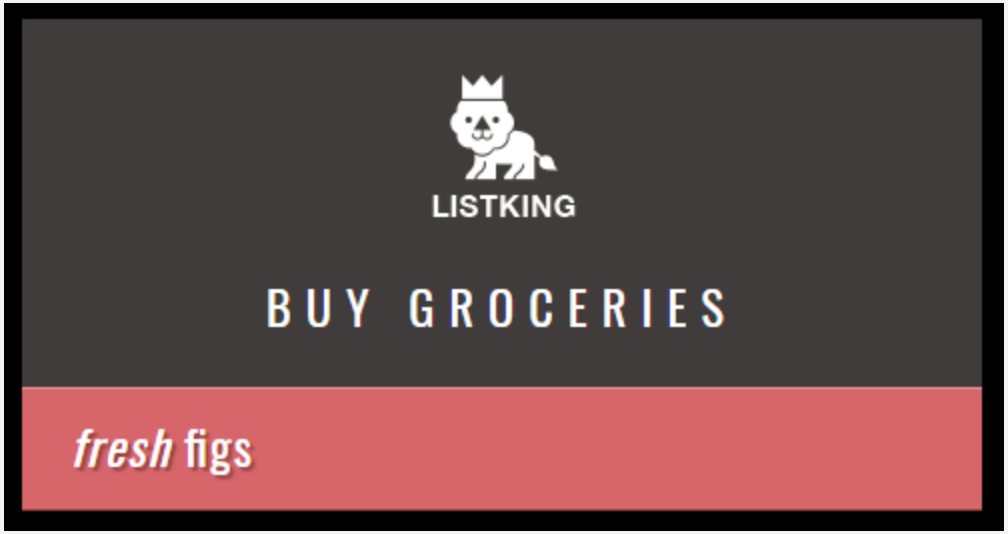
- The animate method can take three optional parameters, as shown here:

```
.animate({
  //styles you want to change
}[,speed][,easing][,complete]);
```

- **speed** indicates the duration of the animation in milliseconds. It can also take the keywords *slow* and *fast*.
- **easing** can have two values: *linear* (the speed of the animation is uniform); or *swing* (speeds up in the middle of the transition, and is slower at the start and end). If no value is specified, swing is used by default.
- **complete** is used to call a function that should run when the animation has finished. This is known as a callback function.

THE ANIMATE METHOD

- In this example, the .animate() method is used to gradually change the values of two CSS properties.
- Both of them have numerical values: opacity and padding-left.



THE ANIMATE METHOD

- All list items are selected and, when a user clicks on one of them, an anonymous function runs.
- Inside it, $(this) creates a new jQuery object holding the element the user clicked on.
- The .animate() method is then called on that jQuery object.

```
$(function() {
  $('li').on('click', function() {
    $(this).animate({
      //styles to be changed
    });
  });
});
```

THE ANIMATE METHOD

- Inside the .animate() method, the opacity and paddingLeft are changed.
- The value of the paddingLeft property is increased by 80 pixels which makes it look like the text is sliding to the right as it fades out.

```
$(function() {
  $('li').on('click', function() {
    $(this).animate({
      opacity: 0.0,
      paddingLeft: '+=80';
    });
  });
});
```

THE ANIMATE METHOD

- The .animate() method has two more parameters.
- The first is the speed of the animation in milliseconds (in this case, 500ms).
- The second is another anonymous function indicating what should happen when the animation finishes. When the animation has finished, the callback function removes that list item from the page using the .remove() method.

```
$(function() {
  $('li').on('click', function() {
    $(this).animate({
      opacity: 0.0,
      paddingLeft: '+=80'
    }, 500, function() {
      $(this).remove();
    });
  });
});
```

JQUERY FORM SELECTORS

- jQuery has selectors that are designed specifically to work with forms.
- However, because of the way jQuery searches for elements, these selectors are not always the quickest way to select elements on a page.
- The jQuery selectors are shown on the next slide.

Selector	
:button	<button> and <input> elements whose type attribute has a value of button.
:checkbox	<input> elements whose type attribute has a value of checkbox.
:checked	Checked elements from checkboxes and radio buttons (see :selected for select boxes)
:focus	Element that currently has focus.
:file	All elements that are file inputs.
:image	All image inputs.
:input	All <button>, <input>, <select> and <textarea> elements.
:password	All password inputs.
:radio	All radio inputs. To select a group of radio buttons, you can use $('input[name="gender"]:radio').
:reset	All inputs that are reset buttons.
:selected	All elements that are selected.
:submit	<button> and <input> elements whose type attribute has a value of submit.
:text	Selects <input> elements with a type attribute whose value is text, or whose type attribute is not present.

JQUERY FORM METHODS

- jQuery also provides methods that can be used with forms. For example, the .val() method gets the value from the first element in a selection; it can also be used to set the value for all matching elements.

Method	Usage
.val()	Primarily used with <input>, <select>, and <textarea> elements. It can be used to get the value of the first element in a matched set, or update the value of all of them
.filter()	Used to filter a jQuery selection using a second selector (especially form-specific filters)
.is()	Often used with filters to check whether a form input is selected/checked
.isNumeric()	Checks whether the value represents a numeric value and returns a Boolean.

JQUERY FORM EVENTS

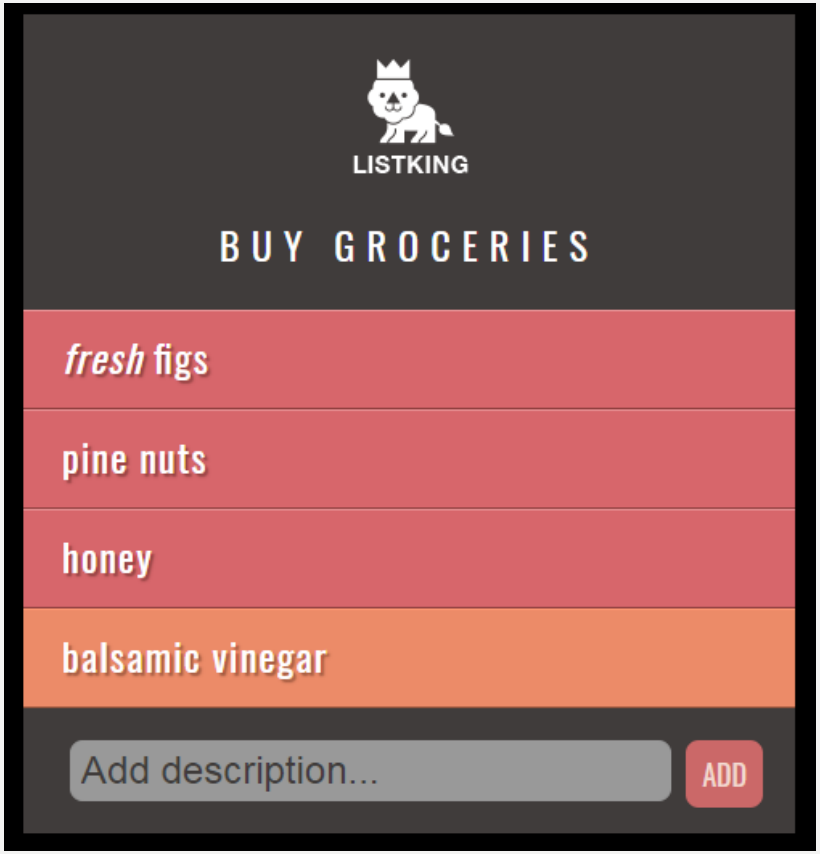
- The events shown here correspond to JavaScript events that you might use to trigger functions. They work with the .on() method; for example:

```
$('#form').on('submit', function(){
  //code to be executed when submit button is clicked
});
```

Method	Usage
blur:	When an element loses focus
change:	When the value of an input changes
focus:	When an element gains focus
select:	When the option for a <select> element is changed
submit:	When a form is submitted

JQUERY FORM EXAMPLE

- In this example, a button and form have been added to the grocery list. When the user clicks on the button to add a new item, the form will come into view.
- The form lets users add a new item to the list with a single text input and a submit button. (The new item button is hidden when the form is in view.)
- When the user presses the submit button, the new item is added to the bottom of the list. (The form is also hidden and the new item button is shown again.)



JQUERY FORM EXAMPLE

- This is the HTML used for the form:

form.html

```
<div id="newItemButton"><button href="#" id="showForm">new item</button></div>
 <form id="newItemForm">
   <input type="text" id="itemDescription" placeholder="Add description..." />
   <input type="submit" id="addButton" value="add" />
 </form>
</div>
```

JQUERY FORM EXAMPLE

- New jQuery objects are created to hold the new item button, the form to add new items, and the add button. These are cached in variables.

```
$(function() {

  var $newItemButton = $('#newItemButton');
  var $newItemForm = $('#newItemForm');
  var $textInput = $('input:text');

});
```

- When the page loads, the CSS hides the new item button (and shows the form), so jQuery methods show the new item button and hide the form.

```
$newItemButton.show();
$newItemForm.hide();
```

JQUERY FORM EXAMPLE

- If a user clicks on the new item button (the <button> element whose id attribute has a value of showForm), the new item button is hidden and the form is shown.

```
$('#showForm').on('click', function(){
  $newItemButton.hide();
  $newItemForm.show();
});
```

JQUERY FORM EXAMPLE

- The last part of the script handles the form submit event.
  - When the form is submitted, an anonymous function is called. It is passed the event object.
  - The .preventDefault() method stops the form from being submitted.
  - The .val() method gets the value the user entered from the text input that is stored in $textInput. The value is stored in a variable called newText.
  - A new item is added to the end of the list using the .after() method.
  - The form is hidden, the new item button is shown again, and the content of the text input is emptied (so the placeholder text is shown again and the user can add a new entry if they want to).

JQUERY FORM EXAMPLE

```
$newItemForm.on('submit', function(e){
  e.preventDefault();
  var newText = $textInput.val();
  $('li:last').after('<li>' + newText + '</li>');
  $newItemForm.hide();
  $newItemButton.show();
  $textInput.val('');
});
```

JQUERY UI

- jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.
- Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.
- https://jqueryui.com/