Introduction to jQuery– Part Three

Website Development 2

Lecture Outline

- $\boxed{\text{jQuery effects}}$
- The .animate() method
- Form elements
- Event Delegation

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

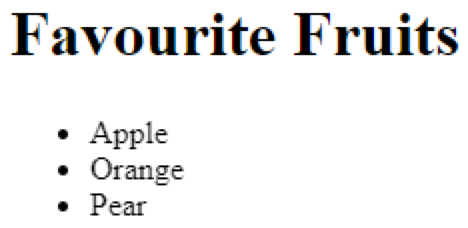
jQuery effects

- 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)

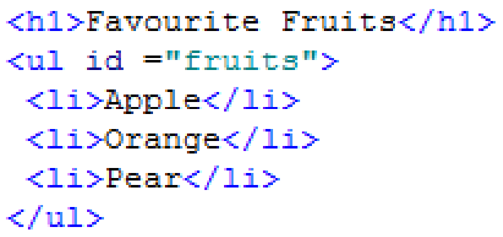
jQuery effects

Fruits Example:

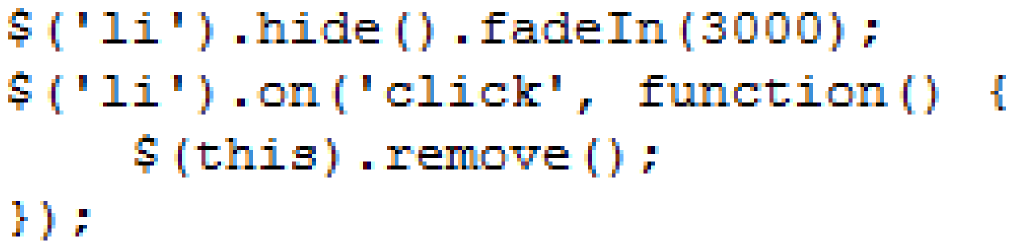


Favourite Fruits

- Apple
- Orange
- Pear

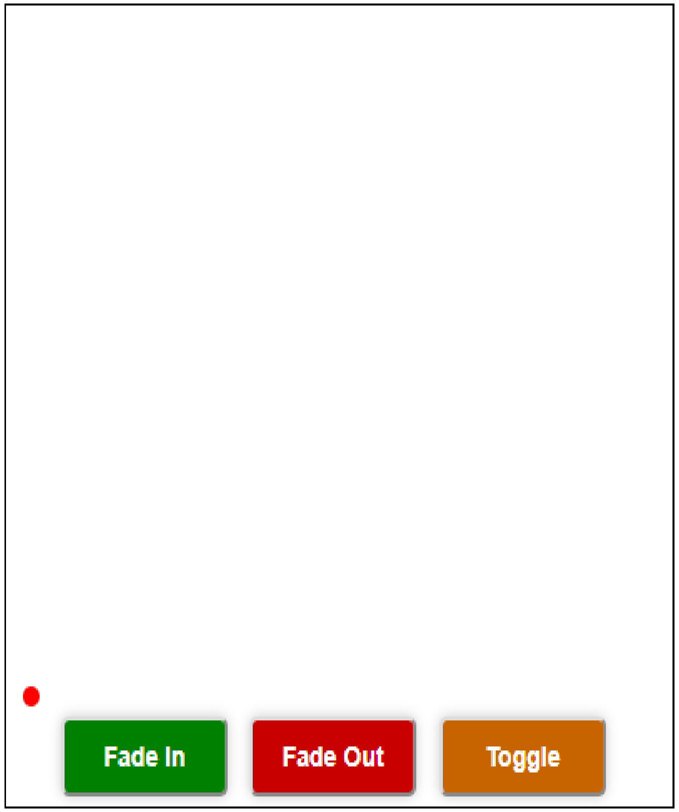


```html
<h1>Favourite Fruits</h1>
<ul id ="fruits">
 <li>Apple</li>
 <li>Orange</li>
 <li>Pear</li>
</ul>
```



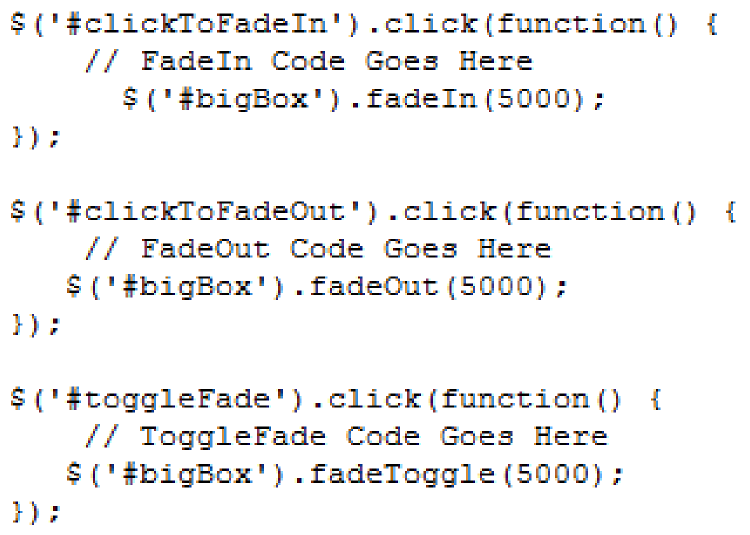
```javascript
$('li').hide().fadeIn(3000);
$('li').on('click', function() {
    $(this).remove();
});
```

jQuery effects



```
<div class="container">
    <div id="bigBox"></div>
    <div class="status"></div>
    <button id="clickToFadeIn">Fade In</button>
    <button id="clickToFadeOut">Fade Out</button>
    <button id="toggleFade">Toggle</button>
</div>
```

jQuery effects



```
$('#clickToFadeIn').click(function() {
    // FadeIn Code Goes Here
      $('#bigBox').fadeIn(5000);
});

$('#clickToFadeOut').click(function() {
    // FadeOut Code Goes Here
   $('#bigBox').fadeOut(5000);
});

$('#toggleFade').click(function() {
    // ToggleFade Code Goes Here
   $('#bigBox').fadeToggle(5000);
});
```

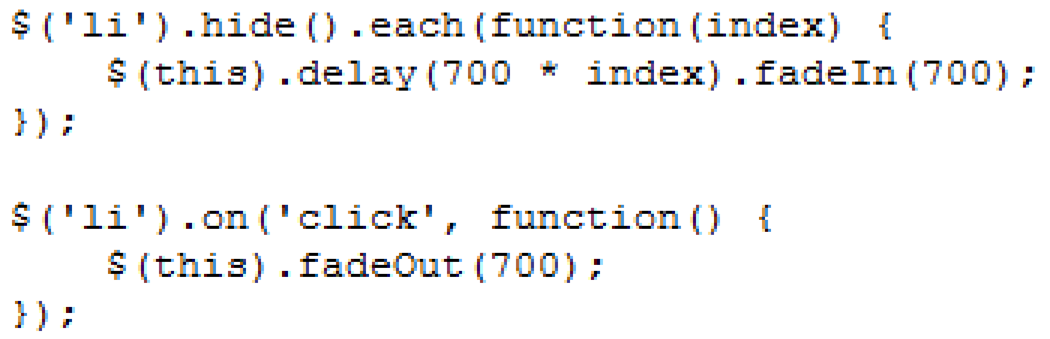
jQuery effects

- 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

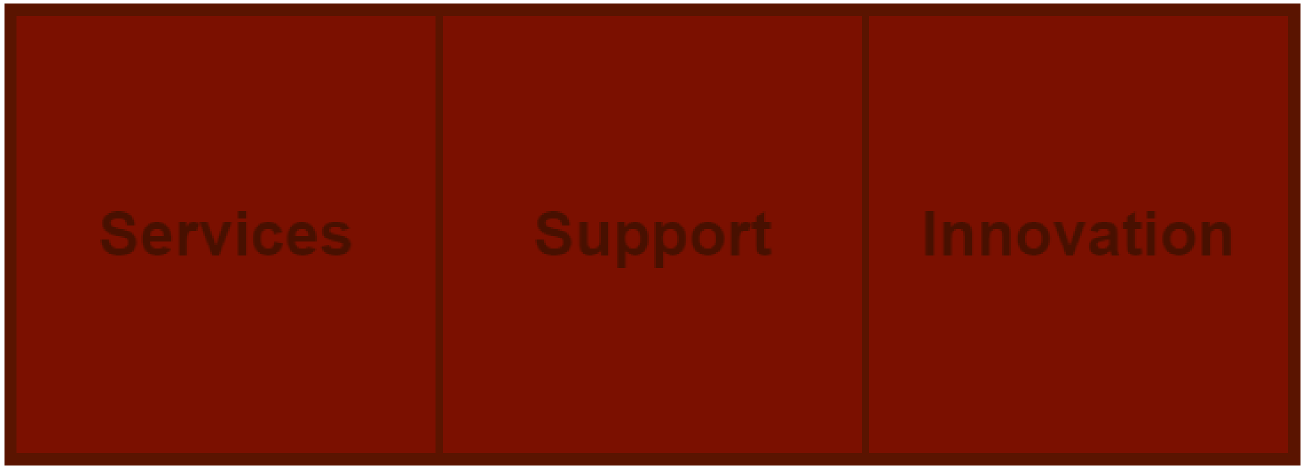
jQuery effects

Fruits Example:



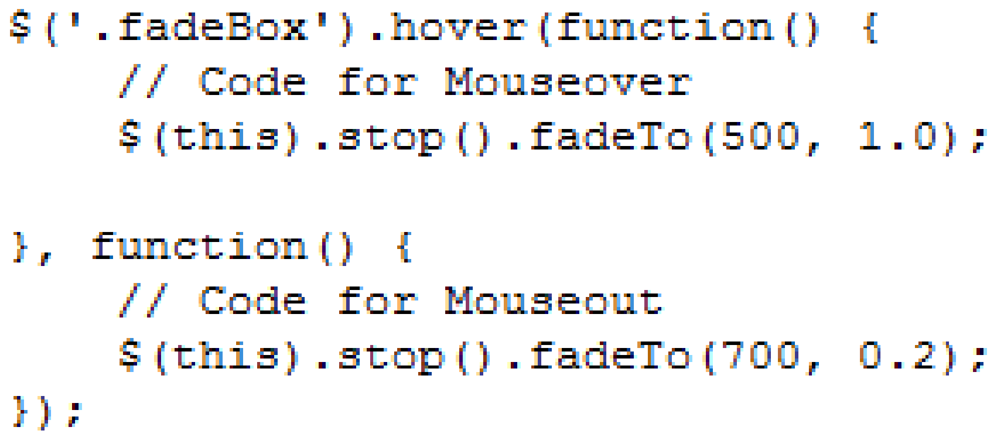
```
$('li').hide().each(function(index) {
    $(this).delay(700 * index).fadeIn(700);
});

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

jQuery effects



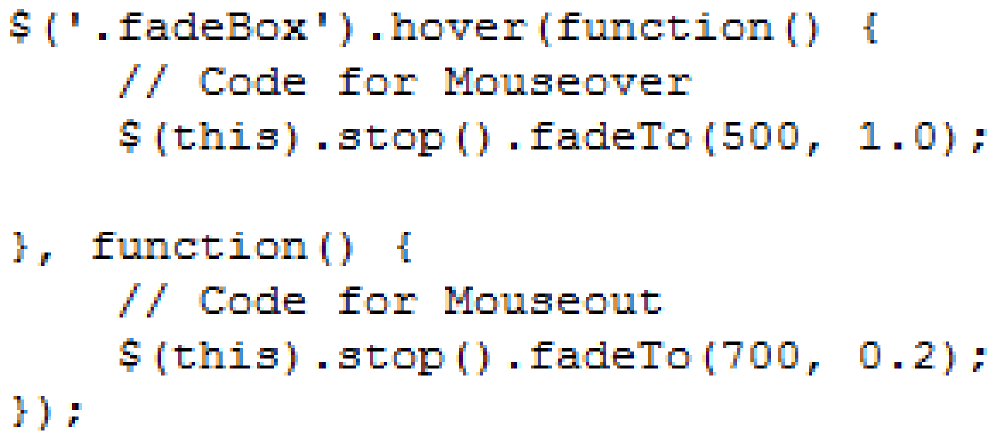
```
<div id="container">
    <div class="fadeBox">
        <h1>Services</h1>
    </div>
     <div class="fadeBox">
        <h1>Support</h1>
    </div>
     <div class="fadeBox">
        <h1>Innovation</h1>
    </div>
</div>
```

jQuery effects

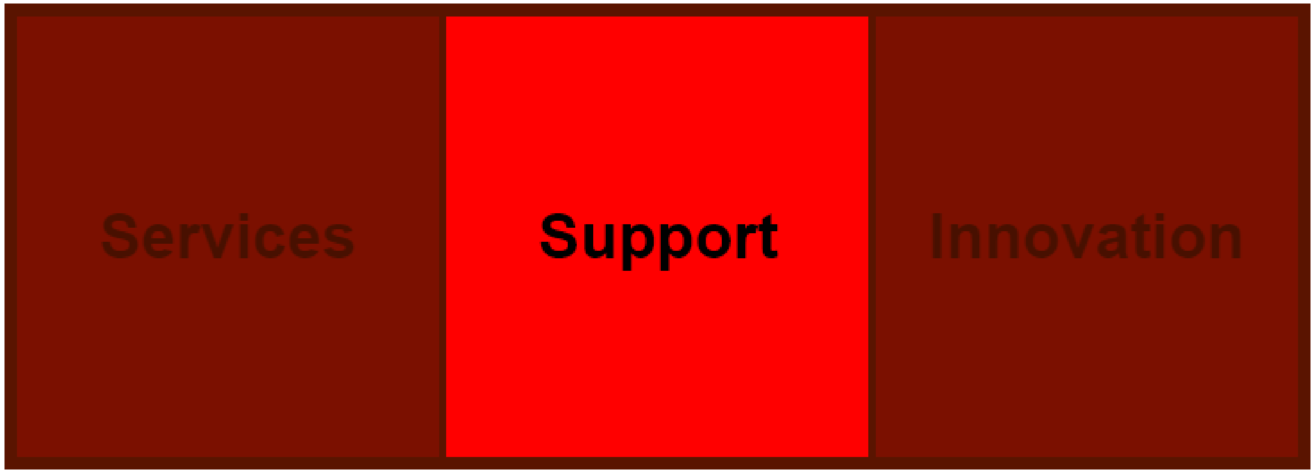


```
$('.fadeBox').hover(function() {
    // Code for Mouseover
    $(this).stop().fadeTo(500, 1.0);

}, function() {
    // Code for Mouseout
    $(this).stop().fadeTo(700, 0.2);
});
```

jQuery effects



```
$('.fadeBox').hover(function() {
    // Code for Mouseover
    $(this).stop().fadeTo(500, 1.0);

}, function() {
    // Code for Mouseout
    $(this).stop().fadeTo(700, 0.2);
});
```



Lecture Outline

- jQuery effects
- $\boxed{\text{The .animate() method}}$
- Form elements
- Event Delegation

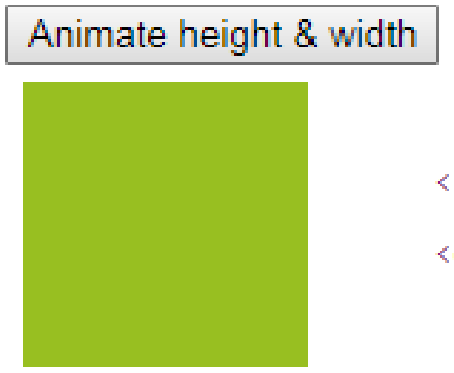
The animate method

- 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:
  - **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

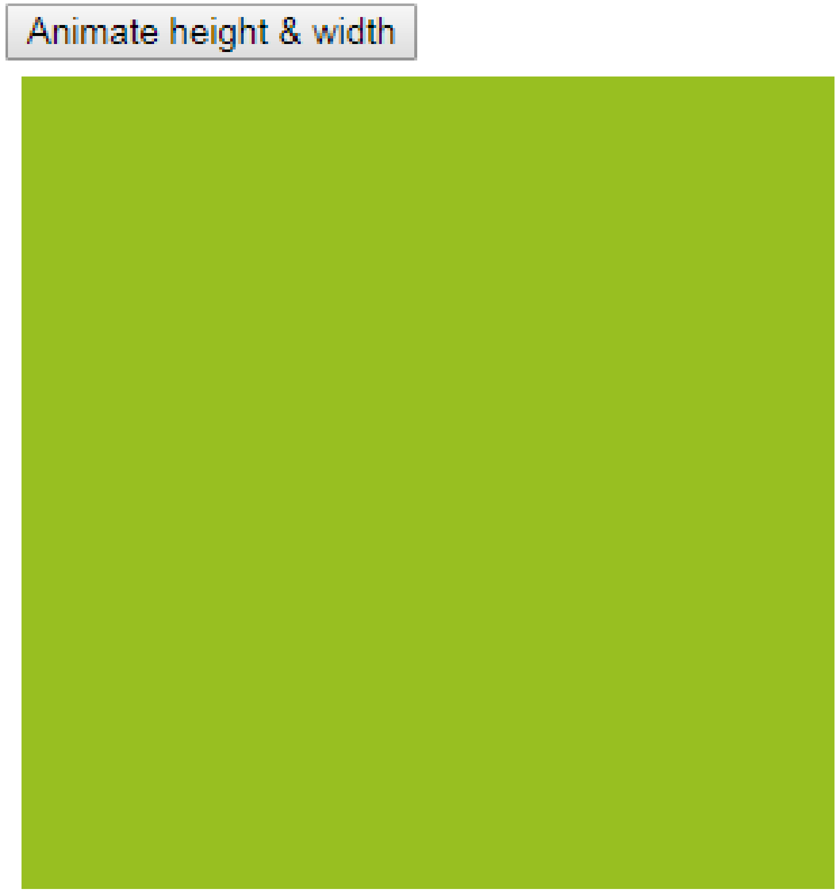


```
<button id="btn">Animate height & width</button>

<div id="box" style="background:#98bf21;height:100px;width:100px;margin:6px;"></div>
```

```
$("#btn").on('click', function(){
  $("#box").animate({
    height: "300px",
    width: "300px"
   }, 5000,"linear");
  });
});
```

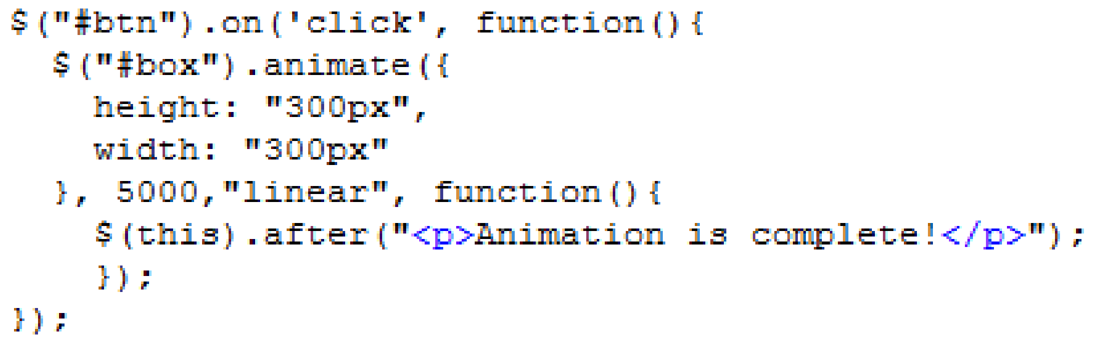
The animate method

- The box has now increased in size to 300px high and wide (from 100px).



The animate method

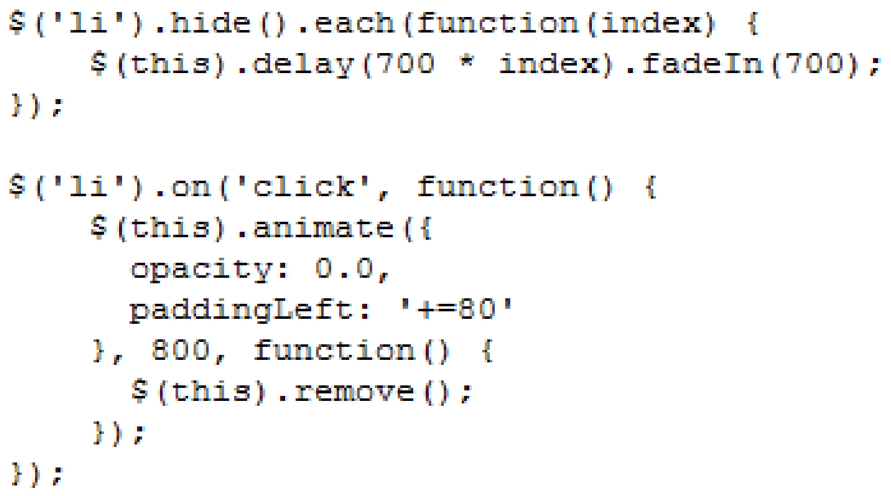
- Recall that one of the parameters for the *animate* method is **complete**, which is used to call a function that should run when the animation has finished. This is known as a callback function.
- If we adjust the code for the previous example as follows:



```
$("#btn").on('click', function(){
  $("#box").animate({
    height: "300px",
    width: "300px"
  }, 5000,"linear", function(){
    $(this).after("<p>Animation is complete!</p>");
    });
});
```

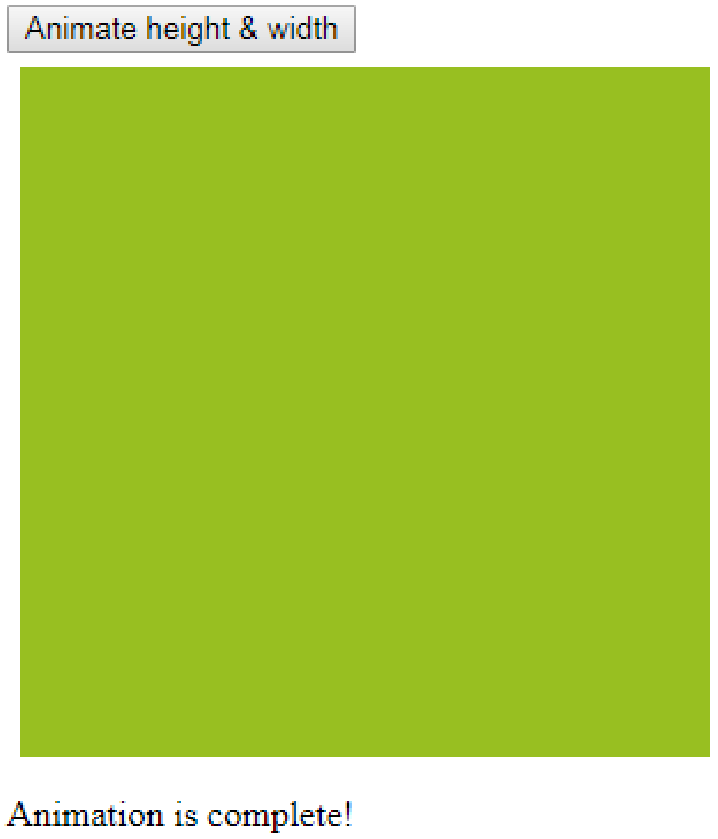
The animate method

Fruits Example:



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

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

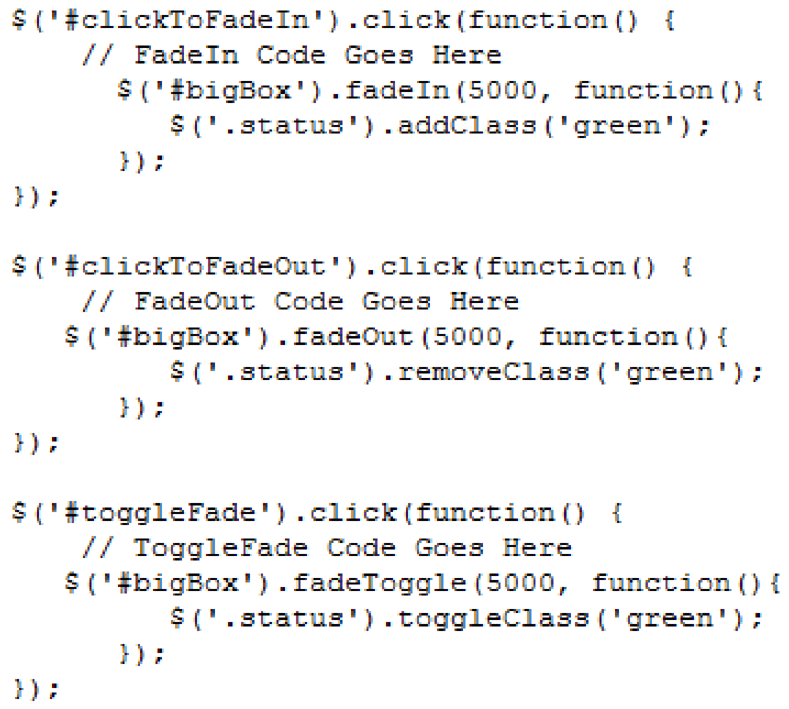
The animate method



Animation is complete!

The animate method

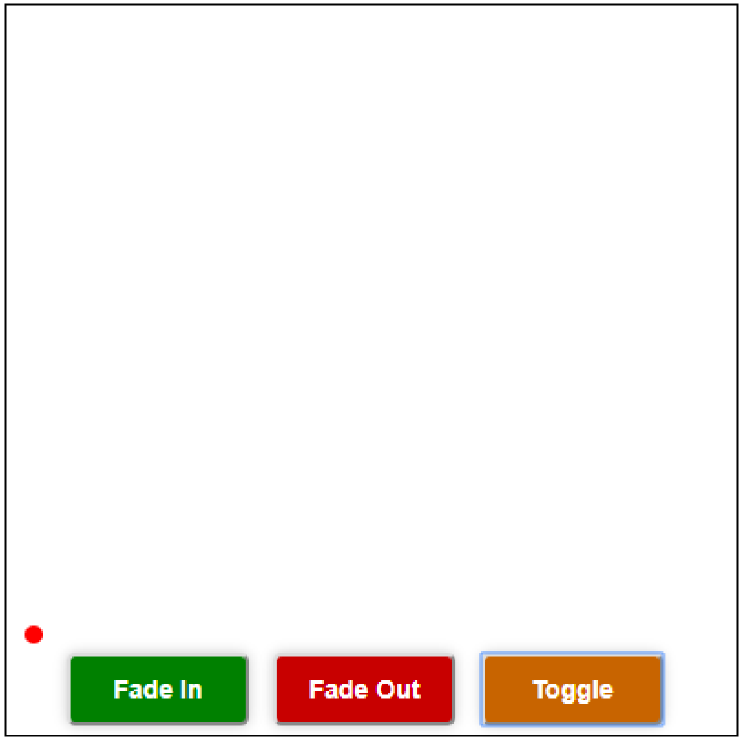
- We will adjust one of our previous ***fade*** examples to use the callback function to change the status colour.

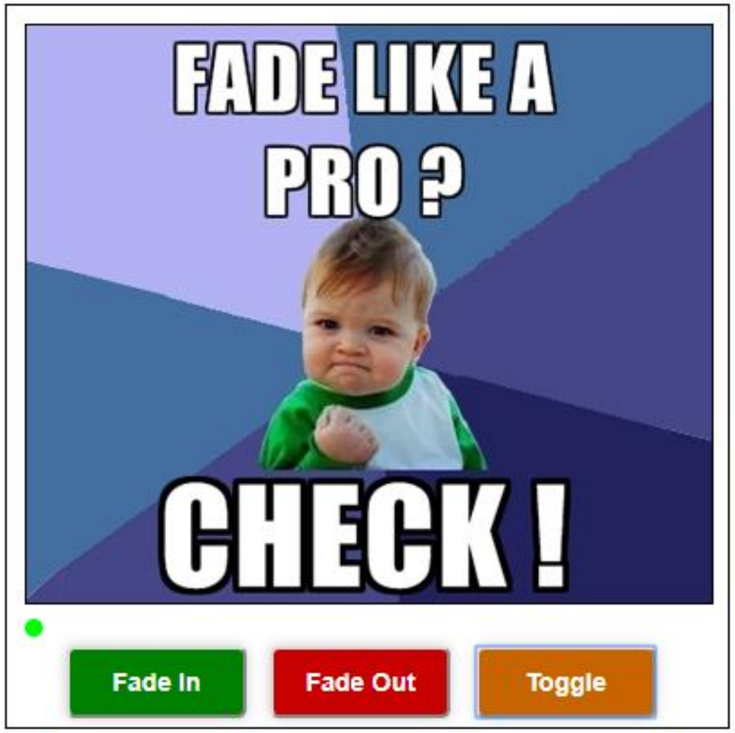


```
$('#clickToFadeIn').click(function() {
    // FadeIn Code Goes Here
      $('#bigBox').fadeIn(5000, function(){
         $('.status').addClass('green');
      });
});

$('#clickToFadeOut').click(function() {
    // FadeOut Code Goes Here
   $('#bigBox').fadeOut(5000, function(){
         $('.status').removeClass('green');
      });
});

$('#toggleFade').click(function() {
    // ToggleFade Code Goes Here
   $('#bigBox').fadeToggle(5000, function(){
         $('.status').toggleClass('green');
      });
});
```

The animate method

- Now note the change in the status colour.





Lecture Outline

- jQuery effects
- The .animate() method
- $\boxed{\text{Form elements}}$
- Event Delegation

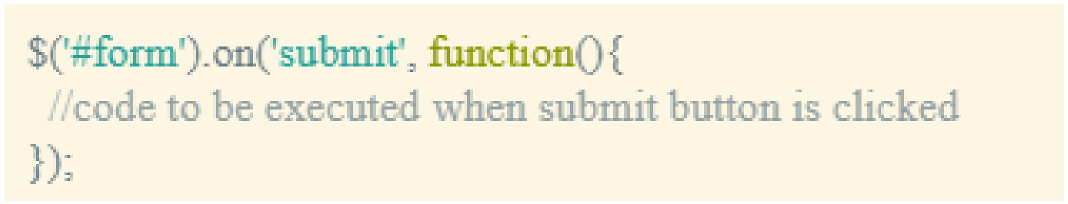
Form elements - methods

- jQuery 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.

Form elements - methods

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

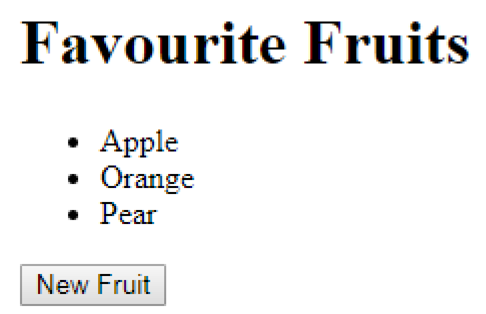


```
$('#form').on('submit', function(){
  //code to be executed when submit button is clicked
});
```

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

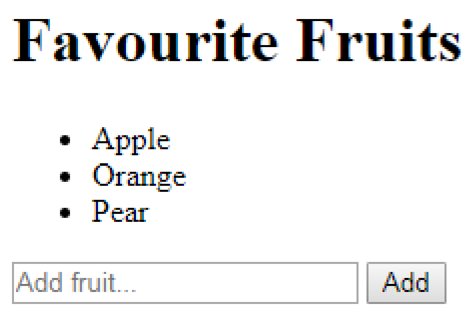
Form elements - methods

Fruits Example:



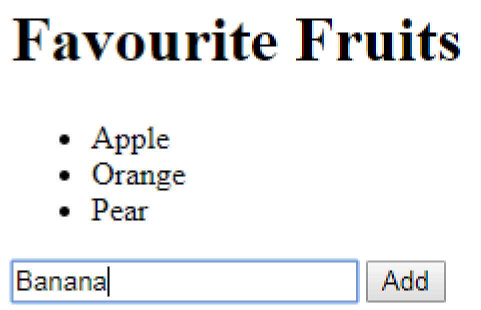
Favourite Fruits

- Apple
- Orange
- Pear



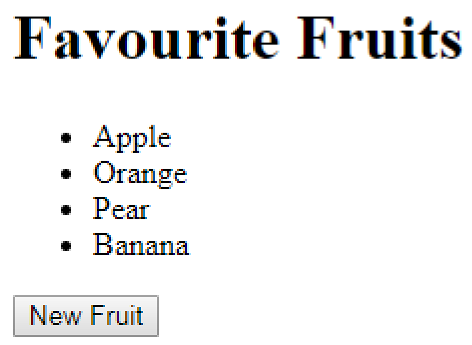
Favourite Fruits

- Apple
- Orange
- Pear



Favourite Fruits

- Apple
- Orange
- Pear

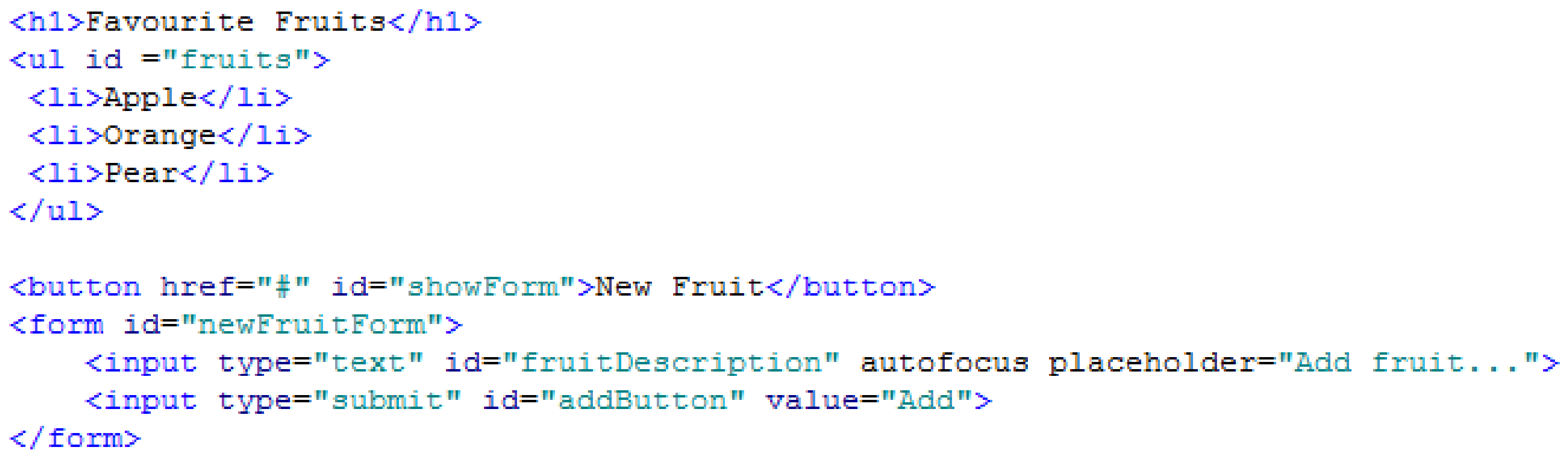


Favourite Fruits

- Apple
- Orange
- Pear
- Banana

Form elements - methods

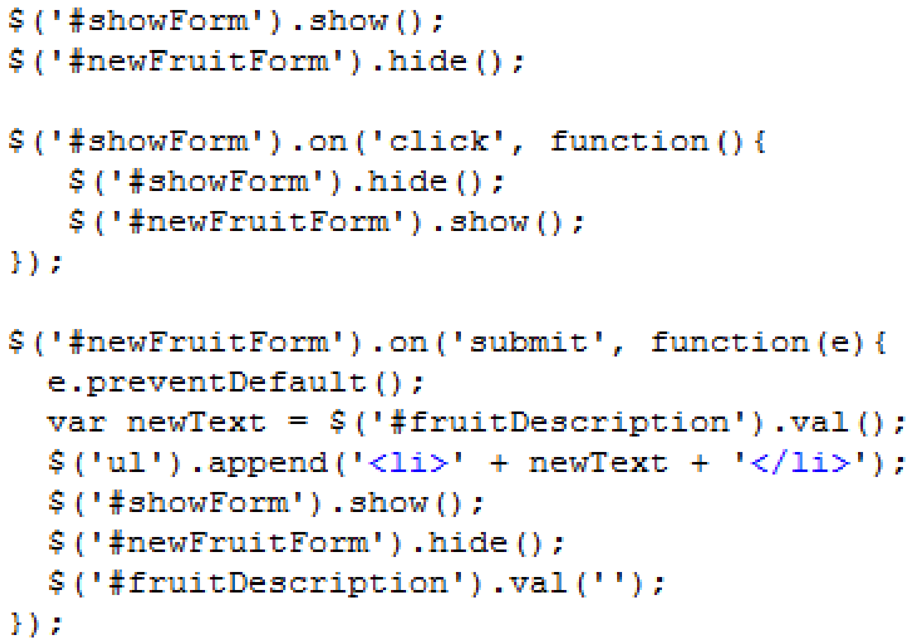
Fruits Example:



```
<h1>Favourite Fruits</h1>
<ul id ="fruits">
 <li>Apple</li>
 <li>Orange</li>
 <li>Pear</li>
</ul>

<button href="#" id="showForm">New Fruit</button>
<form id="newFruitForm">
    <input type="text" id="fruitDescription" autofocus placeholder="Add fruit...">
    <input type="submit" id="addButton" value="Add">
</form>
```

Form elements - methods

Fruits Example:



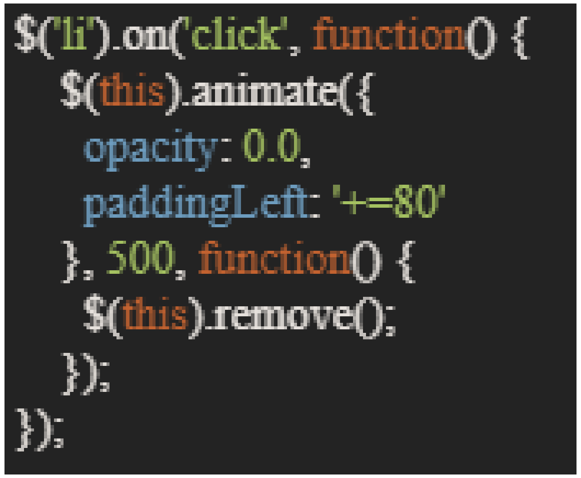
```
$('#showForm').show();
$('#newFruitForm').hide();

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

$('#newFruitForm').on('submit', function(e){
  e.preventDefault();
  var newText = $('#fruitDescription').val();
  $('ul').append('<li>' + newText + '</li>');
  $('#showForm').show();
  $('#newFruitForm').hide();
  $('#fruitDescription').val('');
});
```

Lecture Outline

- jQuery effects
- The .animate() method
- Form elements
- $\boxed{\text{Event Delegation}}$

Event Delegation

- **Event delegation** allows us to attach a single event listener, to a parent element, that will fire for all descendants matching a selector, whether those descendants exist now or are added in the future.
- In our previous slides, we added list items. If we now add in the code to delete the list items as previously:



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

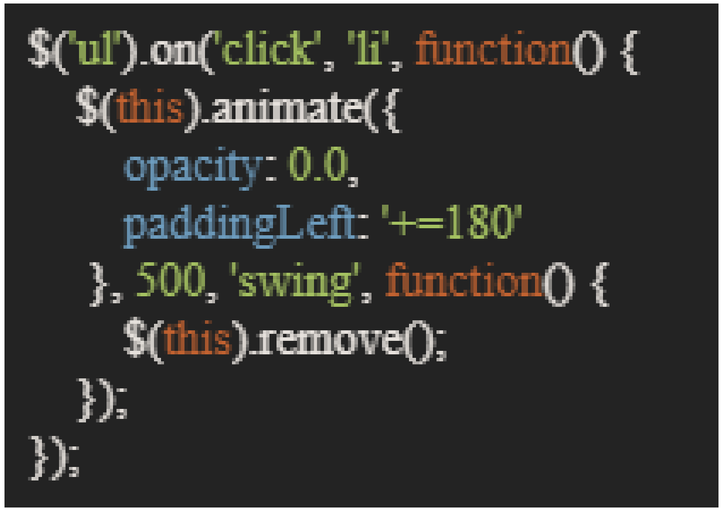
Event Delegation

- This will no longer work (for new elements) because of the directly bound event handler that we attached. Direct events are only attached to elements at the time the associated method is called.

Event Propagation

- Understanding how events propagate is an important factor in being able to leverage Event Delegation. Any time one of our list items are clicked, a click event is fired for that list item, and then bubbles up the DOM tree, triggering each of its parent click event handlers:
    1. <li>
    2. <ul>
    3. <body>
    4. <html>
    5. document root
- This means that anytime you click one of our list item tags, you are effectively clicking over the entire document body! This is called *event bubbling* or *event propagation*. Since we know how events bubble, we can now create a delegated event.

Event Propagation

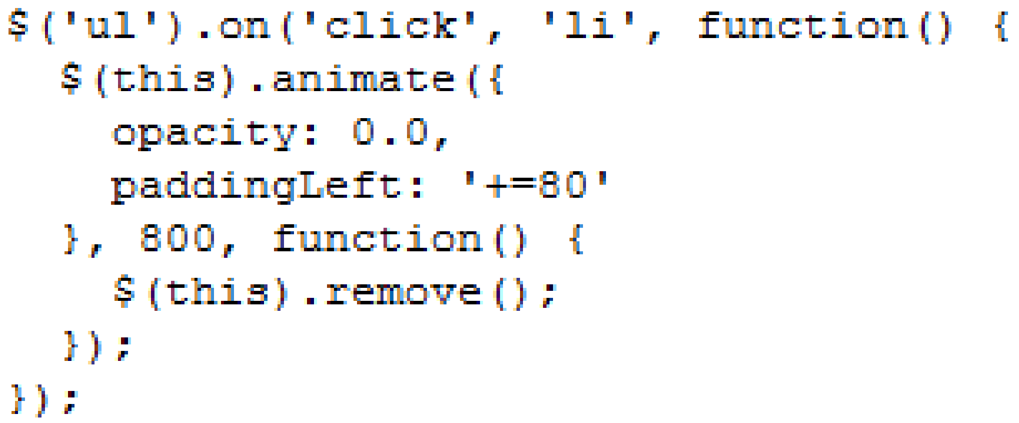


```
$('ul').on('click', 'li', function() {
   $(this).animate({
      opacity: 0.0,
      paddingLeft: '+=180'
    }, 500, 'swing', function() {
      $(this).remove();
   });
});
```

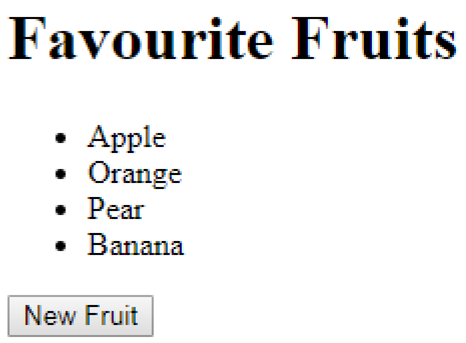
- Notice how we have moved the *li* part from the selector to the second parameter position of the .on() method. This (optional) second, selector parameter tells the handler to listen for the specified event, and when it hears it, check to see if the triggering element for that event matches the second parameter. In this case, the triggering event is our *li* tag, which matches that parameter. Since it matches, our anonymous function will execute. We have now attached event listeners to our <ul> that will listen for *click* events on it's descendant list items, instead of attaching an unknown number of directly bound events to the existing *li* tags only.

Event Propagation

Fruits Example:

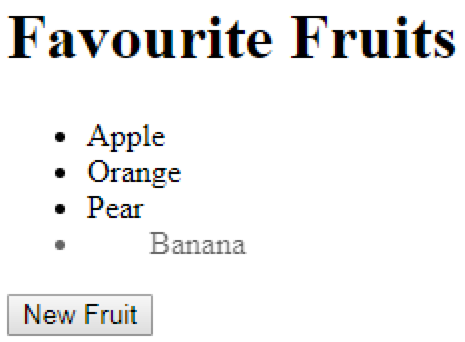


```
$('ul').on('click', 'li', function() {
  $(this).animate({
    opacity: 0.0,
    paddingLeft: '+=80'
  }, 800, function() {
    $(this).remove();
  });
});
```



Favourite Fruits

- Apple
- Orange
- Pear
- Banana



Favourite Fruits

- Apple
- Orange
- Pear
- Banana