

# Web Programming

# **jQuery**

# Web frameworks

- Simplify development
- Code reuse
- Frameworks exist for all components, including
  - JavaScript
  - CSS
  - PHP
  - See: <http://www.bestwebframeworks.com/>

# JavaScript frameworks

- Aims
  - Consistent browser support
  - Simplified, easy-to-use functions
- Number of available options
  - jQuery
  - Prototype
  - Yahoo! UI Library
  - Dojo



- <http://jquery.com>
- JavaScript framework (or library)
  - Probably the most popular one today
- Purpose: make it much easier to use JavaScript
- Simplifies syntax, common actions, DOM manipulation
- Lots of plug-ins and extensions
  - jQuery UI
  - jQuery Mobile
  - ...

# Adding jQuery

- Single JS file to be referenced in HTML <head>
  - Production version (minified and compressed)
  - Development version (uncompressed and readable code)
- Local (downloaded) copy

```
<script src="jquery/2.2.2/jquery.min.js"></script>
```

- Remotely hosted copy from CDN (recommended)
  - CDN = Content Delivery Network (e.g., Google or Microsoft)

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/2.2.2/jquery.min.js"></script>
```

# jQuery syntax

- Remember: it's (still) JavaScript
- Basic syntax: `$("selector").method( )`
- Typical event-based usage:

```
$("selector").event(function () {  
    // perform some action  
});
```

The code that is to be executed when the event is fired is inside a nested function.

- Selector: based on CSS syntax to select DOM elements
- Event: user actions that a web page can respond to
  - click, mouseover, etc...

# The jQuery function \$( )

- Heart and soul of jQuery
- High-level overview
  - Creates a jQuery object
  - Evaluates the expression passed as its parameters
  - Determines how it should respond
  - Modifies itself accordingly

# Outline

- Selectors
- Events
- Methods and actions



# Selectors

# Selectors

- Used to "find" (select) HTML elements based on ids, classes, attributes, etc.
- Based on CSS selectors and syntax
  - Plus some additional selectors
- Complete list:  
[http://www.w3schools.com/jquery/jquery\\_ref\\_selectors.asp](http://www.w3schools.com/jquery/jquery_ref_selectors.asp)

# jQuery Selectors

## Basic selectors

Selector	Example	Selects
<b>element</b>	<code>\$("div")</code>	All <div> elements
<b>.class</b>	<code>\$(".red")</code>	All elements with class="red"
<b>#id</b>	<code>\$("#first")</code>	All elements with id="first"

## Special selectors

Selector	Example	Selects
<b>document</b>	<code>\$(document)</code>	Document
<b>this</b>	<code>\$(this)</code>	Current HTML element

# jQuery Selectors (hierarchy & attr.)

Selector	Example	Selects
<b>parent desc</b>	<code>\$("div p")</code>	All <p> elements that are descendants of a <div> element
<b>parent &gt; child</b>	<code>\$("div &gt; p")</code>	All <p> elements that are a direct child of a <div> element
<b>:parent</b>	<code>\$(":parent")</code>	All elements that are parent of another element
<b>:has(selector)</b>	<code>\$("div:has(p)")</code>	All <div> elements that have a <p> element
<b>:contains(text)</b>	<code>\$(":contains('Hello')")</code>	All elements which contains the text "Hello"
<b>[attribute]</b>	<code>\$("[href]")</code>	All elements with a href attribute
<b>[attribute=val]</b>	<code>\$("[width=10]")</code>	All elements with width=10

# jQuery Selectors (input)

Selector	Example	Selects
<b>:input</b>	<b><code>\$(":input")</code></b>	All input elements
<b>:text</b>	<b><code>\$(":text")</code></b>	All input elements with type="text"
<b>:radio</b>	<b><code>\$(":radio")</code></b>	All input elements with type="radio"
<b>:checkbox</b>	<b><code>\$(":checkbox")</code></b>	All input elements with type="checkbox"
<b>:enabled</b>	<b><code>\$(":enabled")</code></b>	All enabled input elements
<b>:selected</b>	<b><code>\$(":selected")</code></b>	All selected input elements
<b>:checked</b>	<b><code>\$(":checked")</code></b>	All checked input elements

# jQuery Selectors (position-based)

Selector	Example	Selects
<b>:first</b>	<code>\$("p:first")</code>	The first <p> element
<b>:last</b>	<code>\$("p:last")</code>	The last <p> element
<b>:even</b>	<code>\$("tr:even")</code>	All even <tr> elements
<b>:odd</b>	<code>\$("tr:odd")</code>	All odd <tr> elements
<b>:eq(index)</b>	<code>\$("ul li:eq(3)")</code>	The fourth element in a list (index starts at 0)
<b>:lt(no)</b>	<code>\$("ul li:lt(3)")</code>	List elements with an index less than 3
<b>:gt(no)</b>	<code>\$("ul li:gt(3)")</code>	List elements with an index greater than 3

# Example #1 using regular JavaScript

🔗 [examples/jquery/select1\\_js.html](#)

```
<head>
  <script>
    function init() {
      var x = document.getElementsByTagName("p");
      for (var i = 0; i < x.length; i++) {
        x[i].onclick = function() {
          alert(this.innerHTML);
        }
      }
    }
  </script>
</head>

<body onload="init()">
  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Third paragraph</p>
</body>
</html>
```

# Example #1 using jQuery

🔗 [examples/jquery/selector1\\_jquery.html](examples/jquery/selector1_jquery.html)

```
<head>
  <script src="http://ajax.googleapis.com/[...]/jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("p").click(function () {
        alert($(this).html());
      });
    });
  </script>
</head>

<body>
  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Third paragraph</p>
</body>
```



# Example #2

- Now display the alert only for those paragraphs that are (1) within the first div and (2) have the clickable class
- (And changes inside the body of the HTML are **not** allowed!)

```
<body>
<div>
  <p class="clickable">I./1st par</p>
  <p>I./2nd par</p>
  <p class="clickable">I./3rd par</p>
</div>
<div>
  <p>II./1st par</p>
  <p>II./2nd par</p>
  <p>II./3rd par</p>
</div>
</body>
```

# Example #2 using regular JavaScript

🔗 [examples/jquery/select2\\_js.html](#)

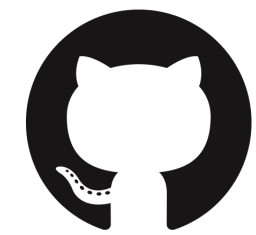
```
<head>
  <script>
    function init() {
      var divs = document.getElementsByTagName("div");
      var x = divs[0].childNodes;
      for (var i = 0; i < x.length; i++) {
        if (x[i].tagName == "P" && x[i].classList.contains("clickable")) {
          x[i].onclick = function () {
            alert(this.innerHTML);
          }
        }
      }
    }
  </script>
</head>
```

# Example #2 using jQuery

🔗 [examples/jquery/select2\\_jquery.html](examples/jquery/select2_jquery.html)

```
<head>
  <script src="http://ajax.googleapis.com/[...]/jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("div:first p.clickable").click(function () {
        alert($(this).html());
      });
    });
  </script>
</head>
```

# Exercise #1



[https://github.com/kbalog/web-programming/tree/master/](https://github.com/kbalog/web-programming/tree/master/exercises/jquery)  
**exercises/jquery**

# Events

# Events

- Most DOM events have an equivalent jQuery method
- Document/window events
  - load(), resize(), scroll(), unload(), ...
- Form events
  - blur(), change(), focus(), select(), submit(), ...
- Keyboard events
  - keydown(), keypress(), keyup(), ...
- Mouse events
  - click(), dblclick(), mouseover(), mouseenter(), mouseleave(), ...

# Waiting for the document to load

- `$(document).ready( )` — execute a function once the document is fully loaded
  - I.e., the jQuery equivalent of `<body onload="init()">`
- Typical jQuery pattern

```
$(document).ready(function() {  
    $("thingToTouch1").event(function() {  
        $("thingToAffect1").method();  
    });  
    $("thingToTouch2").event(function() {  
        $("thingToAffect2").method();  
    });  
    [...]  
});
```

# Example #1

🔗 [examples/jquery/events1.html](#)

```
<head>
  <script src="http://ajax.googleapis.com/.../jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("div").click(function () {
        alert("Clicked a square!");
      });
      $("#red").mouseenter(function () {
        console.log("You entered the red square!");
      });
      $("#red").mouseleave(function () {
        console.log("You left the red square!");
      });
    });
  </script>
</head>

<body>
  <div id="red"></div>
  <div id="blue"></div>
</body>
```



# Event attributes

- Events may have additional attributes
- These can be accessed using the event parameter
- For example:

```
$("input").keydown(function (event) {  
    $("div").html("Key: " + event.which);  
});
```

- See this page for reference:  
[http://www.w3schools.com/jquery/jquery\\_ref\\_events.asp](http://www.w3schools.com/jquery/jquery_ref_events.asp)

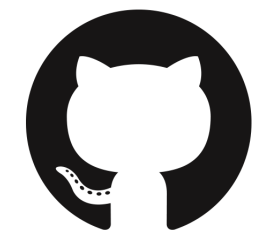
# Example #2

🔗 [examples/jquery/events2.html](#)

```
<head>
  <script src="http://ajax.googleapis.com/.../jquery.min.js"></script>
  <script>
    $(document).mousemove(function (event) {
      $("#pos").html("X: " + event.pageX + ", Y: " + event.pageY);
    });
    $(document).keydown(function (event) {
      $("#pos").html("Key: " + event.which);
    });
  </script>
</head>

<body>
  <p id="pos">Do something</p>
</body>
```

# Exercise #2



[https://github.com/kbalog/web-programming/tree/master/](https://github.com/kbalog/web-programming/tree/master/exercises/jquery)  
**exercises/jquery**

# Attaching/removing event handlers

- Single event

```
$("#p").click(function () {  
    alert("The paragraph was clicked.");  
});
```

- **on()** — attaching (multiple) event handler(s)

```
$("#p").on("click", function () {  
    alert("The paragraph was clicked.");  
});
```

- **off()** — removing event handlers attached with on()

```
$("#p").off("click");
```

# Methods and Actions

# Methods and actions

- Manipulating HTML elements and CSS properties
  - text(), html(), val(), css(), ...
  - [http://www.w3schools.com/jquery/jquery\\_ref\\_html.asp](http://www.w3schools.com/jquery/jquery_ref_html.asp)
- Traversing the DOM
  - parent(), siblings(), first(), last(), ...
  - [http://www.w3schools.com/jquery/jquery\\_ref\\_traversing.asp](http://www.w3schools.com/jquery/jquery_ref_traversing.asp)
- Effects and animation
  - hide(), show(), fadeIn(), slideUp(), ...
  - [http://www.w3schools.com/jquery/jquery\\_ref\\_effects.asp](http://www.w3schools.com/jquery/jquery_ref_effects.asp)

# Calling methods

- Simple method on all matching elements

```
$("#div").addClass("square");
```

- Chaining methods

```
$("#div").show().addClass("square").css("background-color", "red");
```

- Specifying a function to run on each matched element

```
$("#button").click(function(){  
    $("#li").each(function(){  
        alert($("#this").text());  
    });  
});
```

**this** refers to the current element in the iteration

# HTML/CSS methods (1)

## Getting and setting content

Method	Description
<b>text()</b>	Set or return the text content of selected elements
<b>html()</b>	Set or return the content of selected elements (including HTML markup)
<b>val()</b>	Set or return the content of form fields
<b>attr()</b>	Set or return the value of an attribute
<b>prop()</b>	Set or return properties/values of selected elements



# attr() vs prop()

- The difference is important in specific situations, in particular, for input fields

- **attr()** sets/retrieves attributes of an HTML element

- It will equal to the original attribute value

- For example, for this checkbox

```
<input type="checkbox" name="subscribe" checked>
```

- **attr()** will always return the string "checked", even after it is changed by the user

```
$("#input[name=subscribe]").attr("checked"); // "checked"
```

# attr() vs prop()

- **prop()** sets/retrieves property values
- If that value can be changed by the user, use **prop()** instead of **attr()**
- I.e., for inputs, checkboxes, radios, etc. always use **prop()**!
- Example

```
<input type="checkbox" name="subscribe" checked>
```

- **prop()** will return a boolean value true or false

```
$("#input[name=subscribe]").prop("checked"); // true or false
```

# HTML/CSS methods (2)

## Getting and setting CSS classes

Method	Description
<b>css()</b>	Set or return the style attribute
<b>addClass()</b>	Add one or more classes to the selected elements
<b>removeClass()</b>	Remove one or more classes from the selected elements
<b>toggleClass()</b>	Toggle between adding/removing classes from the selected elements

# HTML/CSS methods (3)

## Inserting content

Method	Description
<b>append ( )</b>	Insert content at the end of the selected elements
<b>prepend ( )</b>	Insert content at the beginning of the selected elements
<b>before ( )</b>	Insert content after the selected elements
<b>after ( )</b>	Insert content before the selected elements

## Removing elements/content

Method	Description
<b>remove ( )</b>	Remove the selected element (and its child elements)
<b>empty ( )</b>	Remove the child elements from the selected element

# Creating a new HTML element

🔗 [examples/jquery/create\\_new.html](#)

- Creating new element with HTML

```
var p = "<p>Some text</p>";
```

- Creating new element with DOM

```
var p = document.createElement("p");  
p.innerHTML = "Some text";
```

- Creating new element with jQuery

```
var p = $("<p></p>").text("Some text");
```

- Appending the new element

```
$("body").append(p);
```

# More DOM methods

- Dimensions
  - width(), height(), innerWidth(), outerWidth(), ...
- Traversing up the DOM tree
  - parent(), parents(), parentsUntil()
  - children(), find()
  - siblings(), next(), prev()
- Filtering the selected elements
  - first(), last(), filter(), not()

# Example #1

🔗 [examples/jquery/methods1.html](#)

```
<head>
  <script src="http://ajax.googleapis.com/.../jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("#check").click(function() {
        $("#selected").show().html($("#name").val());
      });
    });
  </script>
</head>

<body>
<form action="">
  <label>Name
    <input type="text" name="name" id="name" size="20"/>
  </label>
  <input type="button" id="check" value="Check"/>
</form>
<p id="selected"></p>
</body>
```

# Example #2

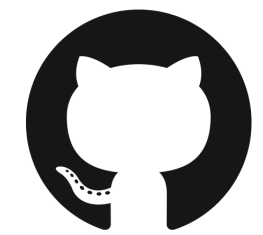
🔗 <examples/jquery/methods2.html>

```
<head>
  <script src="http://ajax.googleapis.com/.../jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("div").addClass("square");
      $("div").mouseenter(function () {
        $(this).css("border", "2px double black");
      });
      $("div").mouseleave(function () {
        $(this).css("border", "1px solid grey");
      });
      $("div").click(function () {
        $(this).toggleClass("red");
      });
    });
  </script>
</head>

<body>
  <div></div>
  <div></div>
```



# Exercise #3



[https://github.com/kbalog/web-programming/tree/master/](https://github.com/kbalog/web-programming/tree/master/exercises/jquery)  
**exercises/jquery**

# Effects

Method	Description
<b>show()</b>	Shows the selected elements
<b>hide()</b>	Hides the selected elements
<b>toggle()</b>	Toggles between the hide() and show() methods
<b>fadeIn()</b>	Set or return the value of an attribute
<b>fadeOut()</b>	Fades out the selected elements
<b>slideUp()</b>	Slides-up (hides) the selected elements
<b>slideDown()</b>	Slides-down (shows) the selected elements
<b>animate()</b>	Runs a custom animation on the selected elements

# Example #3

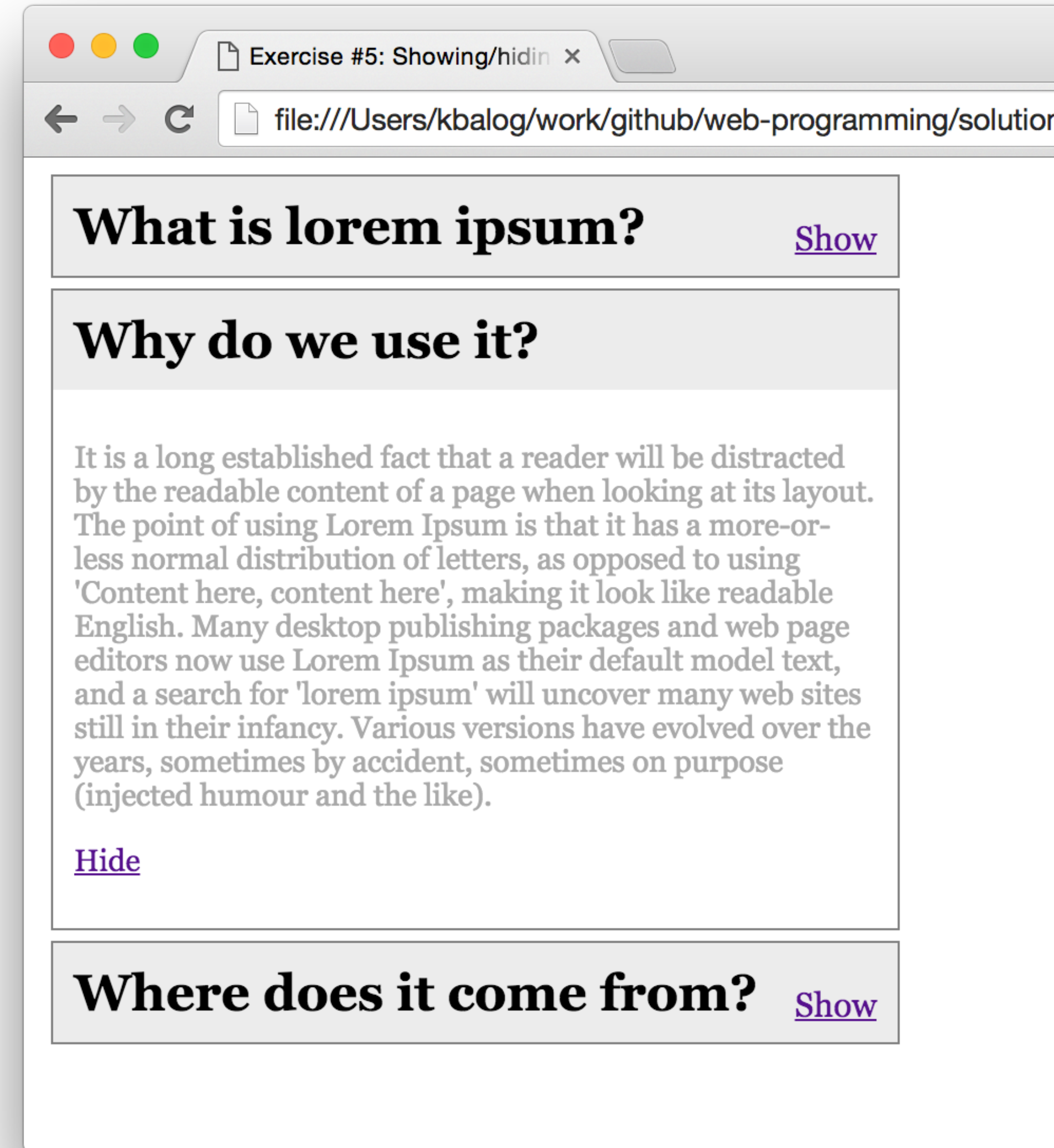
🔗 [examples/jquery/methods3.html](#)

```
<head>
  <script src="http://ajax.googleapis.com/.../jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $("div").mouseenter(function () {
        $(this).fadeOut("slow", 0.15);
      });
      $("div").mouseleave(function () {
        $(this).fadeIn("fast", 1);
      });
    });
  </script>
</head>

<body>
  <div></div>
  <div></div>
  <div></div>
</body>
```

# Example #4

- Exercise #6 from JavaScript Part II.

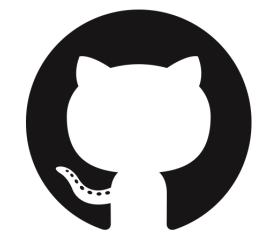


# Example #4

🔗 [examples/jquery/methods4.html](#)

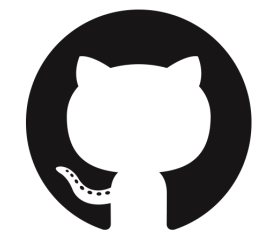
```
$(document).ready(function () {  
    $(".showlink a").click(function () {  
        // hide parent div  
        $(this).parent().hide();  
        // go two levels up, find the element with the "text" class and show it  
        $(this).parent().parent().find(".text").slideDown();  
    });  
  
    $(".hidelink a").click(function () {  
        // hide the element two levels up  
        $(this).parent().parent().slideUp();  
        // show the link three levels up  
        $(this).parent().parent().parent().find(".showlink").show();  
    });  
});
```

# Exercise #4



[https://github.com/kbalog/web-programming/tree/master/](https://github.com/kbalog/web-programming/tree/master/exercises/jquery)  
**exercises/jquery**

# Exercises #5, #6, #7



[https://github.com/kbalog/web-programming/tree/master/](https://github.com/kbalog/web-programming/tree/master/exercises/jquery)  
**exercises/jquery**

# References

- jQuery API Documentation  
<http://api.jquery.com/>
- jQuery Learning Center  
<http://learn.jquery.com/>
- W3Schools jQuery Tutorial  
<http://www.w3schools.com/jquery/>