JAVASCRIPT (CONTINUED)

Statements

```
// expression statement
var x = 1 + 2;

// if/else if/else
// statement
if (condition) {
   statements
}
else if (condition) {
   statements
}
else {
   statements
}
```

```
// while statement
while (condition) {
   statements
}

// for statement
for (init ; test ; inc) {
   statements
}

// function statement
function name (args) {
   statements
}
```

Statements

```
// try catch finally statement
try {
    // normally this code runs from top to bottom
    // sometimes an exception may be thrown
    // either directly with a throw statement,
    // or indirectly by calling another method
}
catch (e) {
    // the statements here are executed if, and only
    // if, the try statement generated an exception
    // these statements handle the exception somehow
}
finally {
    // the statements here are always executed
    // regardless of what happened in the try block
}
```

Objects

- Unordered collections of properties
- Besides the 'dot' operator to access properties, the […] operator can be used
  - one way to think of this is like Python Dictionaries or the Java Hashtable class
- As JavaScript is dynamic and objects can have properties added at any time, this is a very convenient method

```
var cust=new Object();
cust.addr0="36 King St";
cust.addr1="42 Queen Rd";
cust.addr2="16 Abbey St";
```

```
var addr = "";
for (i = 0; i < 3; i++) {
  addr += cust["addr" + i] + '\n';
}
```

Functions

- Functions can also be nested
- Functions support optional arguments
  - if invoked with fewer arguments, undefined is used
- The arguments object can be used with variable length argument lists
  - e.g. Function object has a property **arguments** which can be inspected to find which and how many arguments were given (e.g. if (arguments.length != 3) { … }
- Functions that are properties of objects are usually referred to as **methods**

Classes and Constructors

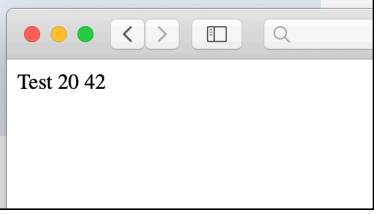
- Creating a class to model Rectangles

```
class Rectangle {
   // Define the constructor
   // Note how it calls a method referred to by "this"
   constructor (idString, widthVal, heightVal) {
      this.id = idString;
      this.resize(widthVal,heightVal);
   }

   // What follows is a method
   resize (widthVal, heightVal) {
      this.width = widthVal;
      this.height = heightVal;
   }
```

Classes and Constructors

- Creating a class to model Rectangles (cont)

```
    // Here is another method
    getArea () {
        return this.width * this.height;
    }
}

// Test out the constructor and methods
var rect = new Rectangle ("Test", 4, 5);
document.writeln(rect.id);
document.writeln(rect.getArea());
rect.resize(6, 7);
document.writeln(rect.getArea());
```



Regular Expressions

- A regular expression is an object that describes a pattern of characters that can be used to perform **pattern matching** and **search** and **replace** actions on text
- Often RegExps can be thought of as programs within a program
- However, despite their utility, they can be a documentation nightmare

Some people, when confronted with a problem, think
"I know, I'll use regular expressions."
Now they have two problems.
J. Zawinski, '97

Regular Expressions

- In JavaScript, regular expressions are represented by RegExp objects
- Syntax of a regular expression:
  - */pattern/modifiers*;
  - Example 1:

    ```
    var re1 = /Free/i;
    ```

    - "Free" is a pattern and "i" is a modifier (case insensitive)
  - Example 2:

    ```
    var re2 = /s$/;
    ```

    - match any string that ends with 's'

Regular Expressions

- RegExp object methods:
  - search() – returns starting position of the first match, or -1
    - Example:

      var str = "Visit W3Schools";
      var n = str.search(/w3schools/i);
      returns 6
  - exec() – returns the first match, or null
    - Example:

      var str = "Visit W3Schools";
      var match = /w3schools/i.exec(str);
      returns "W3Schools"
  - test() – returns true if there is a match, false otherwise
    - Example:

      var str = "Visit W3Schools";
      var match = /w3schools/i.test(str);
      returns "true"

More details:
https://www.w3schools.com/jsref/jsref_obj_regexp.asp

The Problem

- Despite JavaScript and DOM being functionally useful, coding on the client-side is not particularly easy
- Consider Java and its huge standard library of useful functionality
- DOM scripting entails a lot of repetitive domain-specific boilerplate coding

A Solution

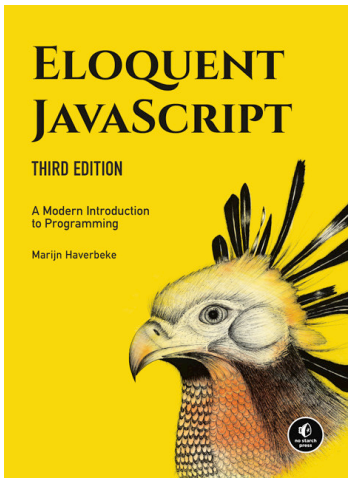
- JavaScript needs its own standard library
- Focus on the domain-specific programming tasks
- More than one solution: *jQuery, YUI, MooTools…*

Other Resources

W3Schools JavaScript Tutorial
http://www.w3schools.com/js/default.asp

Or search for **Douglas Crockford** & **JavaScript** for some video lectures by JavaScript's cranky evangelist:

- The World's Most Misunderstood Language
- The Good Parts
- 4 Part Series for Beginners
- Advanced JavaScript



ELOQUENT JAVASCRIPT

http://eloquentjavascript.net

University of Glasgow

jQuery

Web Application Development 2

Better JavaScript with jQuery
Selecting, Decorating, Enhancing

BETTER JAVASCRIPT WITH JQUERY

"WRITE LESS, DO MORE"

jQuery

- Created by John Resig
- One of the most popular JS libraries
- Simplifies client-side scripting:
  - selecting DOM elements
  - creating UI animations and effects
  - handling events
  - developing AJAX applications

Cross Browser Compatibility

- jQuery takes a lot of the problems out of developing for multiple browsers
- Acts as a layer of abstraction over various browsers
- No more browser sniffing

Plug-in Architecture

- jQuery creates a useful foundation for additional functionality to be added
- A wide range of specialised plug-ins have been developed since the release of jQuery for all manner of web-dev tasks (e.g., jQueryUI)

jQuery Syntax $()

- jQuery uses a basic pattern of **selecting** and **acting** on a particular DOM element and manipulating its parameters
- The selectors of CSS are reused in jQuery

```
$('#name')   .text   ('the new text');
$('p')       .css    ('color', 'blue');
```

Select | Action | Parameters

Clean, Consistent Markup

- jQuery reuses that pattern
- Heavy use of anonymous functions
- Chaining functions together

Ensure page 'ready'

```
$(document).ready(function() {
  alert('Hello World!');
});
```

- (document).ready ensures code inside isn't executed until page has loaded
- Shorthand: $()

```
$(function() {
  alert('Shorter form!');
});
```

jQuery Events

Mouse Events	Keyboard Events	Form Events	Document / Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
hover			

- Syntax:

```
$('p')  .click  (function() {…});
```

Select Event Action

Attaching Event Handlers

- Here we toggle some text between visible (show) and invisible (hide)

```
$('#toggleButton').click(function() {
  if ($('#disclaimer').is(':visible')) {
    $('#disclaimer').hide();
  } else {
    $('#disclaimer').show();
  }
});
```

Attaching Event Handlers

```
<html>
  <head>
    <script src="jquery.js"></script>
    <script type="text/javascript">

      $(document).ready(function() {
        $("a").click(function() {
          alert("Hello World!");
        });
      });
    </script>
  </head>

  <body>
    <a href="">1st Link</a><br><br>
    <a href="">2nd Link</a>
  </body>
</html>
```

Change Colour

```
<head>
  <script src="jquery.js"></script>
  <style> p#text {color : red;} </style>
  <script type="text/javascript">

    $(document).ready(function() {
      $("#toggleButton").click(function() {
        $("p#text").css(
                  {fontSize:36, color:"blue"});
      });
    });

  </script>
</head>

<body>
  <button id="toggleButton"> Click here </button>
  <p id="text">This text will change colour</p>
</body>
```

Show / Hide

```
<head>
   <script src="jquery.js"></script>
   <script type="text/javascript">

     $(document).ready(function() {
       $("#toggleButton").click(function() {
         $("p#disclaimer").css("color","blue");
         if ($('#disclaimer').is(':visible')) {
            $('#disclaimer').hide();
         } else {
            $('#disclaimer').show();
         }
       });
     });

   </script>
</head>
<body>
  <button id="toggleButton">Click here</button>
  <p id="disclaimer">A standard disclaimer.</p>
</body>
```

Animating Elements

```
<html>
  <head>
    <script src="jquery.js"></script>
    <script type="text/javascript">

     $(document).ready(function() {
        $(".bigtext").hover(function() {
          $(this).animate({paddingLeft: '+=30px'}, 200);
        }, function() {
          $(this).animate({paddingLeft: '-=30px'}, 200);
        });
      });

    </script>
    <style type="text/css" media="screen">
      .bigtext { font-size: 400%; }
    </style>
  </head>

  <body>
    <a href="">1st Link</a><br><br>
    <a href="" class="bigtext">2nd Link</a>
  </body>
</html>
```

Multiple Events

- Use the on() method

```
$('p')      .on      ({ click : function() {…}, … });
```

Select | on | Event: Action

```
$("p").on({
   mouseenter: function(){
     $(this).css("background-color", "red");
   },
   mouseleave: function(){
     $(this).css("background-color", "green");
   },
   click: function(){
     $(this).css("background-color", "yellow");
   }
});
```

JQUERY VALIDATION
#PLUGIN DEMO

BEFORE WITH JAVASCRIPT

```
<body>
    <div id="page”>
      <div id="header”>
        <h1 id="title">PuppyIR: BaSe (Basic
             Search)</h1>
      </div> <!-- end header -->

      <div id="searchbox”>
        <form action="/base/query/"
              onsubmit="return validate_form(this)"
              method="post">

          {% csrf_token %}

          <input type="text” name="query"
                 value="" id="query">

          <input type="submit" value="Search" />

        </form>
      </div> <!-- searchbox -->
```

Form Validation

Form validation

```
<script type="text/javascript">

   function validate_form(thisform) {
      with (thisform) {
         if (query.value==null || query.value=="") {
            query.focus();
            return false;
         }
         else {
            return true;
         }
      }
   }

</script>
```

AFTER WITH JQUERY

jQuery Form Validation

```
<head>
  <meta http-equiv="Content-type" content="text/html; >
  <title>PuppyIR: BaSe2 (Basic Search Improved)</title>
  <link href="{{ MEDIA_URL }}base2/css/base.css"
        rel="stylesheet" title="basestyle"
        type="text/css">
  <script type="text/javascript"
          src="{{ MEDIA_URL }}base2/jquery/jquery.js">
  </script>
  <script type="text/javascript"
          src="{{ MEDIA_URL }}base2/jquery/validate.js">
  </script>

  <script type="text/javascript">
    $(document).ready(function() {
      $("#myform").validate();
    });
  </script>
</head>

<body>
  <!-- snip -->
</body>
```

Animation Demo

```
<head> <script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({left: '100px'}, "slow");
        div.animate({fontSize: '3em'}, "slow");
        div.animate({height: '200px'}, "slow");
        div.animate({left: '8px'}, "slow");
        div.animate({fontSize: '1em'}, "slow");
        div.animate({height: '100px'}, "slow");
    });
});
</script> </head>

<body>
<button>Start Animation</button>
<p>By default, all HTML elements have a static position,
and cannot be moved. To manipulate the position,
remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21; height:100px; width:
200px; position:absolute;">HELLO</div>
</body>
```

Append to DOM

```
<head> <script>
$(function(){
    $("#btn1").click(function(){
        $("p").append(" <b>Appended text</b>.");
    });

    $("#btn2").click(function(){
        $("ol").append("<li><b>Appended item</b></li>");
    });
});
</script> </head>

<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>
</body>
```

More Demos

- Demo: jQuery UI
  http://jqueryui.com/
- Interactions:
  http://jqueryui.com/demos/draggable/
- Widgets:
  https://jqueryui.com/button/
- Effects:
  http://jqueryui.com/demos/show/

JQUERY FUNDAMENTALS

http://jqfundamentals.com