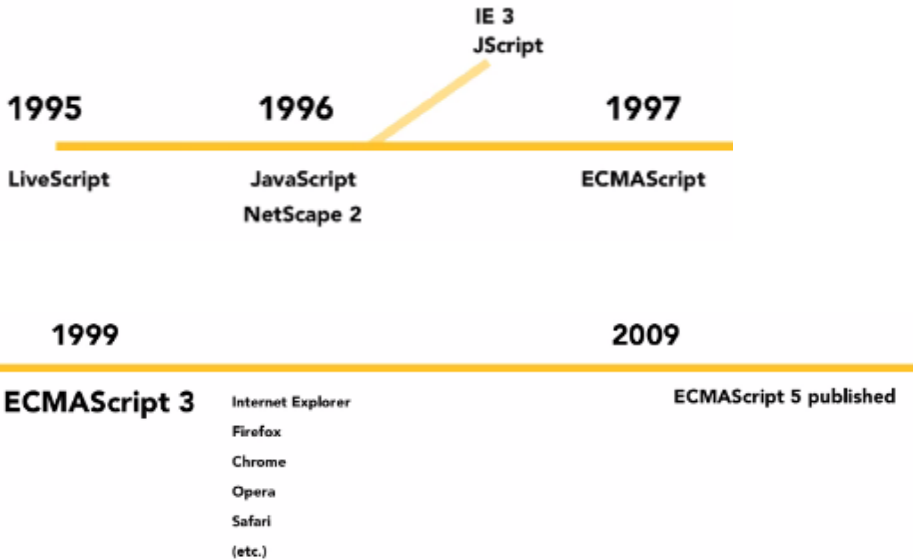
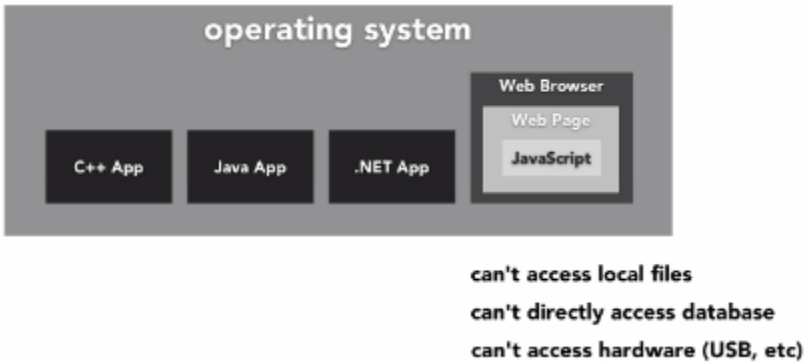


# JavaScript Essentials 2013

## Introduction

### What is JavaScript?

- Scripting language for the web
- No client-side IO features
- Not compiled
- Case Sensitive



## Chapter 1: JavaScript General Conventions

- Case sensitive
- Use semicolon for end statement
- Comments
  - `//`
  - `/* */`
- Browser executes code as soon as it sees it

## JavaScript Location

- Create function files for all the code logic
- Attach to the html pages as head or use inline calls as you need them

- Fully qualified `<script>` tag

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

- HTML5 assumes `<script>` tag is javascript

```
<script src="myscript.js">
</script>
```

- Have the `<script>` tag at the very bottom before `</body>` tag

## Chapter 2: JavaScript Components and Syntax

### Variables

- These create space in memory and assigned as 'undefined'

```
var year;
var customerEmail;
var todaysDate;
```

- Creates space in memory AND assigns it a value of '2011'

```
var year = 2011;
```

### Allowed Characters

- letters, numbers, `_`, `$`

## If statement

```
if ( condition ) {
    // code goes here
    //
}

if ( condition ) {
    // code goes here
    //
} else {
    // code goes here
    //
}
```

## Code Block { }

```
If (somethingIsTrue){
// code block
}

If (somethingIsTrue)
{
// code block
}

If (somethingIsTrue){ // code block }
```

## Operators

### Assignment

```
=
```

### Arithmetic

```
+ - * /
+= -= *= /=
```

### Increment/Decrement (unary operator)

```
++
--
a++
b--
--a
--b
```

### Prefix

```
var a = 5;  
alert(++a)
```

increment a, then run alert()

## PostFix

```
var a = 5;  
alert(a++)
```

run alert(), then increment a

## Ternary

```
var result = condition ? true : false  
  
var player12 = 500;  
var player2 = 600;  
  
var highScore;  
  
highScore = (playerOne > playerTwo) ? playerOne : playerTwo;
```

## Modulus ( remainder )

```
%
```

## Operator Precedence

```
* /  
+ -  
(( 1 + 1 ) / 2)
```

## Comparison Operators

```
== <= >= !=
```

## Logical AND/OR

```
&& AND  
|| OR
```

## Assignment and Equality

```
= assignment  
==equality  
=== strict equality
```

## Loops

### While Loop

```
a=1;
while (condition){

a++;
}
```

## Do..While Loop

```
do{

a==
} while (condition);
```

## For Loop

```
for(var a=1; a<100; a++){
  //runs 99 times
}
```

## Break

```
for (var a=1; a<100; a++){

  if( imNotDone ){
    // go back to the iteration
    // without doing remaining code for iteration
    continue;
  }

  if( imDone ){
    break; // out of loop
  }
}
```

## Console


### console.log


- Can change icons in firebug to aid in debugging



.log  
.debug  
.info  
.warn  
.error

they are not equal

they are not equal

 they are not equal

 they are not equal

  they are not equal

## Functions

- Best to declare them BEFORE calling them

```
function createMessage(parameters){  
    // code to run  
    return mySomethingVar;  
}
```

### Parameter Mismatch

- Extra parameters will be ignored
- Too few parameters causes others to be 'undefined'

### Variable Scope

```
var globalVar;  
  
function tempVars(){  
    var localVar;  
}
```

## Chapter 3: Arrays and Objects

### Arrays

```
var myArray = [];  
var myArray = [1,2,3,"mouse",:tree",6]  
var myArray = new Array(); //object oriented version  
var myArray = Array(); //object oriented version  
var myArray = Array(5); // 5 slots in the array  
  
var myTwoDimensionalArray = [,];
```

### Array Properties

.length

### Array Methods

.reverse() – creates a new array

.join()

.sort()

### *Find all Anchor tags on a page*

```
document.getElementsByTagName("a");
```

## Numbers

- All JavaScript Numbers are 64-bit floating point numbers

### Addition vs. Concatentation

Both Numbers: Arithmetic

```
var foo = 5;
var bar = 5;
console.log(foo+bar); //10
```

### Both Strings: Concatenate

```
var foo = "5";
var bar = "5";
console.log(foo+bar); //55
```

### Mixed: Treat as String

```
var foo = 5;
var bar = "5";
console.log(foo+bar); //55 - one is a string
```

### Mixed: Some operations will generate NaN

```
var foo = 5;
var bar = "5";
console.log(foo*bar); // NaN - not a number
```

Can use this to check for validation of forms:

```
var foo = "55";
var mytest = Number(foo); // make it a Number, NaN if it can't
if (!isNaN(mytest)){
  // do something
}
```

## Strings

### Quotes Inside Quotes

```
'Don't mix your quotes' BAD
"Don't mix your quotes" GOOD
"He said,"Bye" and left" BAD
"He said,\"Bye\" and left" GOOD
```

### String Properties

```
.length
```

### String Methods

```
.toUpperCase()
.toLowerCase()
.split(",") // slit at comma
```

`.indexOf("string")` // # or -1 if not found

`.slice(6,5)` // grab char 7 and take 5 chars


< and > can be used for alphabetical

## STRING COMPARISON

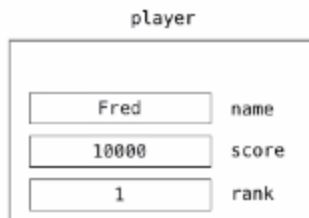
```
var str1 = "aardvark";  
var str2 = "beluga";  
  
if (str1 < str2 ) { ...    // true  
  
var str1 = "aardvark";  
var str2 = "Beluga";  
  
if (str1 < str2 ) { ...    // false!  
  
ABCD... less than  abcd...
```

## Object Orientation

```
var playerName = "Fred";  
var playerScore = 10000;  
var playerRank = 1;
```

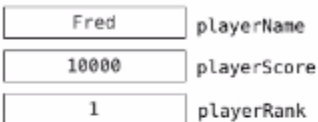


```
var player = new Object();  
player.name = "Fred";  
player.score = 10000;  
player.rank = 1;
```



```
var playerName = "Fred";  
var playerScore = 10000;  
var playerRank = 1;
```

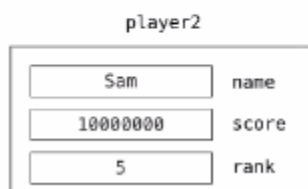
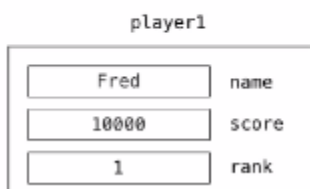
**variables**



```
var player = new Object();  
player.name = "Fred";  
player.score = 10000;  
player.rank = 1;
```



```
var player1 = { name: "Fred", score: 10000, rank: 1 };  
var player2 = { name: "Sam", score: 10000000, rank: 5 };
```





## Object Orientation in Use

```
var player1 = {name: "Fred", score: 100000, rank:1};
var player2 = {name: "Sam", score: 19999, rank:2};

function playerDetails(){
    // display info on the players

    // if we know this function will be associated
    with an object..
    console.log(this.name + " has a rank of: " +
this.rank +
                                " and a score of " +
this.score);
}

...

player1.logDetails = playerDetails;
player2.logDetails = playerDetails;

player1.logDetails();
player2.logDetails();
```

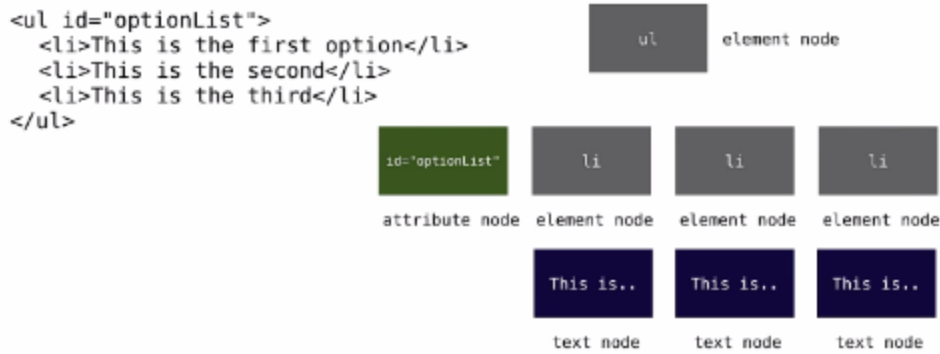
```
Fred has a rank of: 1 and a score of 100000
Sam has a rank of: 2 and a score of 19999
```

## Chapter 4: DOM

### Document Object Model

document.

## ELEMENT, ATTRIBUTE AND TEXT NODES



## Element nodes don't contain text



## HOW TO GET AN ELEMENT NODE

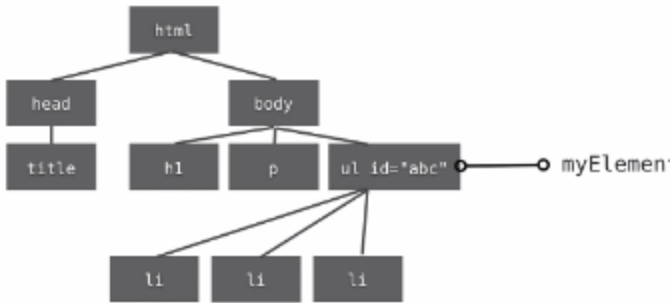


## If element has an id...

`document.getElementById("someId");`

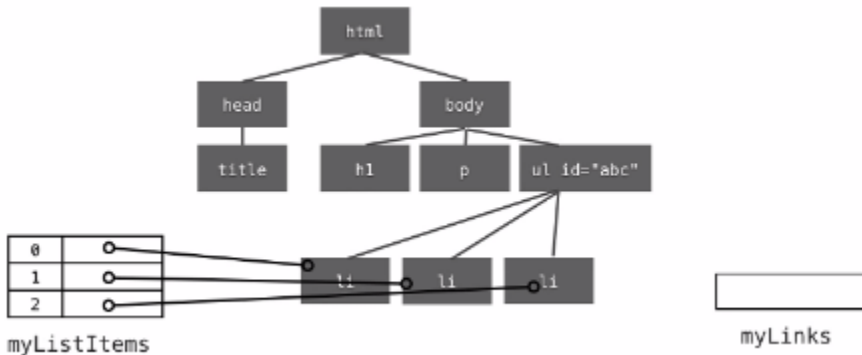
```
var myElement = document.getElementById("abc");
```

```
var myElement = document.getElementById("abc");
```



ByTagName ( Creates an Array)

```
document.getElementsByTagName("a");
```



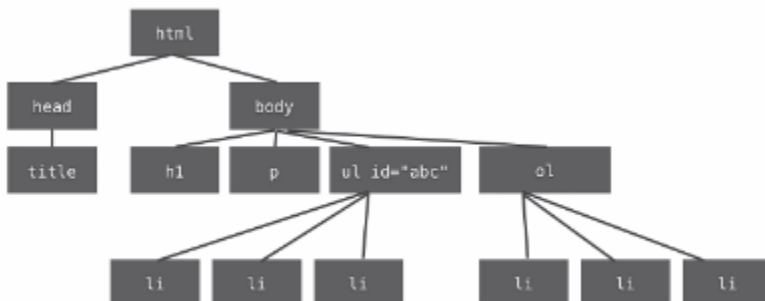
## Getting Elements by Id or TagName

```
var mainTitle = document.getElementById("mainTitle");
console.log("This is an element of type: ", mainTitle.nodeType );
console.log("the Inner HTML is ", mainTitle.innerHTML);
console.log("Child nodes: ", mainTitle.childNodes);
```

```
var myLinks = document.getElementsByTagName("a");
console.log("Links: ", myLinks.length);
```

## RESTRICTING ELEMENTS TO RETRIEVE

```
var myListItems = document.getElementsByTagName("li");
var myFirstList = document.getElementById("abc");
var limitedList = myFirstList.getElementsByTagName("li");
```



## Changing Element Attributes

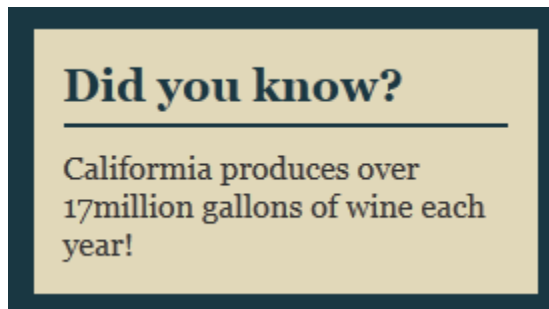
```
var mainContent = document.getElementById("mainContent");
mainContent.setAttribute("align", "right");

var mainTitle = document.getElementById("mainTitle");
console.log(mainTitle.innerHTML);

var sidebar = document.getElementById("sidebar");
console.log(sidebar.innerHTML);
```

## Creating DOM Content

### Creating Elements



### .innerHTML Method

```
var newHeading = document.createElement("h1");
var newParagraph =
document.createElement("p");

newHeading.innerHTML = "Did you know?";
newParagraph.innerHTML = "California produces
over 17million gallons of wine each year!";

document.getElementById("trivia").appendChild(
newHeading);
document.getElementById("trivia").appendChild(
newParagraph);
```

## Better to use `TextNode` to assign text values

```
// create new elements
var newHeading = document.createElement("h1");
var newParagraph = document.createElement("p");

// create associated text nodes....
var h1Text = document.createTextNode("Did you know?");
var paraText = document.createTextNode("California produces over 17million gallons of
wine each year!");

// append, instead of create...
newHeading.appendChild(h1Text);
newParagraph.appendChild(paraText);

// append to page DOM..
document.getElementById("trivia").appendChild(newHeading);
document.getElementById("trivia").appendChild(newParagraph);
```

```
//Alternatives to APPENDCHILD
// create new element BEFORE THE END
parent.insertBefore(newElement, existingElement);
var myNewElement = docueme.createElement("li");
var secondItem = myElement.getElementsByTagName("li")[1];
myElement.insertBefore(myNewElement, secondItem);
```

## EVENTS and EVENT LISTENERS

Examples, normally..

- lowercase
- start with 'on'
  - onload
  - onclick
  - onmouseover
  - onblur
  - onfocus

```
// #1 INLINE scripting
<button onclick="alert('Hello, world');" >
  Run Some JavaScript
</button>
```

## *Use semicolon because whole thing is a STATEMENT*

```
element.event = function();
```

```
// #2 ANONYMOUS FUNCTION
myElement.onclick = function(){
  // event handler code ...
  // ...
  // ...
};
```

```
// #3 using DOM addEventListener...
// (event sans 'on', functionname, false)
document.addEventListener('click', myFunction, false);
document.addEventListener('click', anotherFunction, false);
document.removeEventListener('click', anotherFunction, false);

//Browsers inconsistent..IE8 and previous
document.attachEvent('onclick', myFunction);
```

## CROSS-BROWSER ADD EVENT HELPER METHODS

---

```
function addCrossBrowserEventListener (elementName, eventName,
functionName) {
  // does the addEventListener function exist?
  if (elementName.addEventListener) {
    // yes - use it
    elementName.addEventListener(eventName, functionName, false);
    return true;
  } else {
    // otherwise use attachEvent
    elementName.attachEvent("on" + eventName, functionName);
    return true;
  }
}
```

Use premade script for this....

Use JQuery (or another library) instead of writing you own cross-browser code

## Common Events

```
document.onclick = function() {
  alert("You clicked somewhere
in the document");
}

document.onclick = function() {
  alert("You clicked somewhere
in the document");
}
var myImage =
document.getElementById("mainIm
age");
myImage.onclick = function(){
  alert("You clicked the
image");
}
```

### window.onload

- only write the window.onload function ONCE
- ONLY the last declared function will be used if there are multiple ones

```
function prepareEventHandlers() {  
    var myImage = document.getElementById("mainImage");  
    myImage.onclick = function() {  
        alert("You clicked the image");  
    }  
}  
  
window.onload = function() {  
    //prep anything we need to  
    prepareEventHandlers();  
}
```

## onfocus & onblur

```
var emailField = document.getElementById("email");
```

```
emailField.onfocus = function() {  
    if ( emailField.value == "your email" ) {  
        emailField.value = "";  
    }  
};
```

```
emailField.onblur = function() {  
    if ( emailField.value == "" ) {  
        emailField.value = "your email";  
    }  
};
```

## Timers

Assign a timeout to an INTERVAL, then  
setTimeout(function,millisconds)  
setInterval(function,millseconds)  
clearInterval(intervalHandle)

```
// Two methods for timers - setTimeout and setInterval (single / repeating)

function simpleMessage() {
    alert("This is just an alert box");
}

// setTimeout is in milliseconds
//setTimeout(simpleMessage,5000);

var myImage = document.getElementById("mainImage");

var imageArray = ["_images/overlook.jpg","_images/winery_sign.jpg","_images/lunch.jpg",
"_images/bigSur.jpg","_images/flag_photo.jpg","_images/mission_look.jpg"];
var imageIndex = 0;

function changeImage() {
    myImage.setAttribute("src",imageArray[imageIndex]);
    imageIndex++;
    if (imageIndex >= imageArray.length) {
        imageIndex = 0;
    }
}

// setInterval is also in milliseconds
var intervalHandle = setInterval(changeImage,5000);

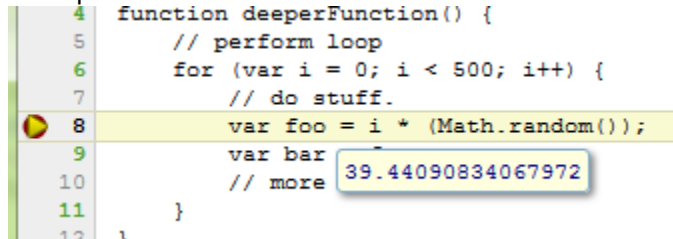
myImage.onclick = function() {
    clearInterval(intervalHandle);
};
```

## FireBug

### Debugging controls

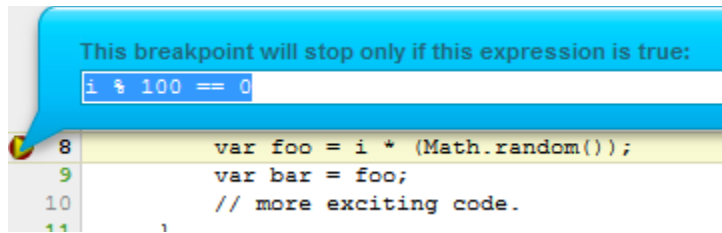


### Breakpoint: Gutter: Left-Click



### Set breakpoint by Expression: Gutter Right Click





## Forms

field values  
field events

form events

## Validating Forms

If forms have id or name, can use DOM to access

### GETTING FORM AND FORM ELEMENTS

```
<form id="frmContact" name="frmContact" method="post" action="thanks.htm">
  <fieldset id="personalInfo">
    <legend><strong>Personal Information</strong></legend>
    <p>
      <label for="name">Name</label>
      <input name="name" type="text" class="text" id="name" tabindex="100"
    </p>
    <p>
      <label for="email">Email</label>
      <input name="email" type="text" class="text" id="email" tabindex="11
    </p>
    <p>
      <label for="phone">Phone</label>
      <input name="phone" type="text" class="text" id="phone" tabindex="13
    </p>
  </fieldset>
</form>
```

Document

## TEXTFIELDS

### main property

`myTextField.value`

### main events

`onfocus`   `onkeypress`  
`onblur`   `onkeydown`  
`onchange`   `onkeyup`

## CHECKBOXES AND RADIO BUTTONS

### main property

`myCheckBox.checked`  
(true or false)

☒ Please send me monthly discounts!   ☐ No thanks!

### main events

`onclick`   `onchange`

## SELECT

### main properties

`mySelect.type`  
(select-one or select-multiple)

select-one

`mySelect.selectedIndex`

select-multiple

`mySelect.options[x].selected`  
(true or false)

### main events

`onchange`

lynda.com

## FORM EVENTS

Validate form data entry by catching the `onsubmit`

**General Information**

I would like more information about the following tours

<input type="checkbox"/> Backpack Cal	<input type="checkbox"/> Cycle California	<input type="checkbox"/> Nature Watch
<input type="checkbox"/> California Calm	<input type="checkbox"/> From Desert to Sea	<input type="checkbox"/> Snowboard Cal
<input type="checkbox"/> California Hot Springs	<input type="checkbox"/> Kids California	<input type="checkbox"/> Taste of California

Comments

Would you like to receive our monthly tour specials newsletter? No, we don't spam or share your information with anyone else. [Read our privacy policy.](#)

☒ Please send me monthly discounts! ☐ No thanks!

**main event**

○ **onsubmit**

**return false;**

## onsubmit Return FALSE

- Return False on the 'onsubmit' event will prevent form submission; validate here

```
// handle the form submit event
function prepareEventHandlers() {
    document.getElementById("frmContact").onsubmit = function() {
        // prevent a form from submitting if no email.
        if (document.getElementById("email").value == "") {
            document.getElementById("errorMessage").innerHTML = "Please provide at
least an email address!";
            // to STOP the form from submitting
            return false;
        } else {
            // reset and allow the form to submit
            document.getElementById("errorMessage").innerHTML = "";
            return true;
        }
    };
}

// when the document loads
window.onload = function() {
    prepareEventHandlers();
};
```

## Show Hide Components

```
// show and hide sections of a form
function preparePage() {
    document.getElementById("brochures").onclick = function() {
        if (document.getElementById("brochures").checked) {
            // use CSS style to show it
            document.getElementById("tourSelection").style.display = "block";
        } else {
            // hide the div
            document.getElementById("tourSelection").style.display = "none";
        }
    };
    // only hide the form element IF JAVASCRIPT is ENABLED...
    // now hide it on the initial page load.
    document.getElementById("tourSelection").style.display = "none";
}

window.onload = function() {
    preparePage();
};
```

Check here to receive brochures on our tours: ☐

Send

Check here to receive brochures on our tours: ☒

- |  |   |  |
|--|---|--|
| <input type="checkbox"/> Backpack Cal        | <input type="checkbox"/> Cycle California   | <input type="checkbox"/> Nature Watch        |
| <input type="checkbox"/> California Calm     | <input type="checkbox"/> From Desert to Sea | <input type="checkbox"/> Snowboard Cali      |
| <input type="checkbox"/> California Hotspots | <input type="checkbox"/> Kids California    | <input type="checkbox"/> Taste of California |

Send

## Setting Inline Styles

```
myElement.style.color = "#FF0000";
myElement.style.left = "40px";
myElement.style.backgroundRepeat = "repeat-x";
```

## Style Property Naming

- Properties with dashes, convert to camelCase
- Always set the values as a string
- Semicolon to end property assignment

```
#example {  
  width: 230px;           myElement.style.width = "230px";  
  color: #fff;            myElement.style.color = "#fff";  
  font-weight: bold;      myElement.style.fontWeight = "bold";  
  background-color: #193742; myElement.style.backgroundColor = "#193742";  
}
```

## Applying Classes

“class” is a reserved word in JavaScript

```
myElement.class  
myElement.className = "someCSSclass";  
myElement.className = "";
```

- If an Element is assigned a certain class, replace it with....
- Tighter/closer matched CSS will override styling here...

```
// prevent a form from submitting  
function preparePage() {  
  document.getElementById("mainContent").onclick = function() {  
    if ( document.getElementById("mainContent").className == "example") {  
      document.getElementById("mainContent").className = "";  
    } else {  
      document.getElementById("mainContent").className = "example";  
    }  
  };  
}  
  
window.onload = function() {  
  preparePage();  
};
```

## Moving CSS styled items based off a timer

```
var currentPos = 0;
var intervalHandle;

function beginAnimate() {
    document.getElementById("join").style.position = "absolute";
    document.getElementById("join").style.left = "0px";
    document.getElementById("join").style.top = "100px";
    // cause the animateBox function to be called
    intervalHandle = setInterval(animateBox,50);
}

function animateBox() {
    // set new position
    currentPos+=5;
    document.getElementById("join").style.left = currentPos + "px";
    //
    if ( currentPos > 900) {
        // clear interval
        clearInterval(intervalHandle);
        // reset custom inline styles
        document.getElementById("join").style.position = "";
        document.getElementById("join").style.left = "";
        document.getElementById("join").style.top = "";
    }
}

window.onload = function() {
    setTimeout(beginAnimate,5000);
};
```

## Conventions

## Variables and Functions

```
var score;
var highScore;
var evenHigherScore;
function calculate(){...
function calculateDistance(){...
function checkFormFields(){...
```

## Objects:

- Math
- Date

```
var myDate = new Date();
```

## Brace Style ( Always Use Blocks { } )

```
if (x) {  
    //...  
    //...  
} else {  
    //...  
    //...  
}
```

## Define Functions BEFORE You Call Them

```
function animateBox() {  
    // set new position  
    if ( currentPos > 900) {  
        // clear interval  
        // reset custom inline styles  
    }  
}  
  
window.onload = function() {  
    setTimeout(beginAnimate,5000);  
};
```

## Search Javascript style guidelines

### Javascript Minification

- SAVES SPACE
- Compressed to save space and EOL errors
- Replace long var and function names
- Tools
  - JSMIn
  - YUI Compressor
  - Google Closure Compiler

### Simple

The screenshot shows the Closure Compiler web interface in a browser. The URL is <http://closure-compiler.appspot.com/home>. The interface includes a text area for input code, a "Compile" button, and a "Reset" button. The output area shows the compiled code, which is minified. The status bar indicates "Compilation was a success!".

Original Size: 1.12KB (517 bytes gzipped)  
Compiled Size: 595 bytes (249 bytes gzipped)  
Saved 47.94% off the original size (51.84% off the gzipped size)  
The code may also be accessed at [default.js](#).

Compiled Code

```
var currentPos=0, intervalHandle;function beginAnimate(){document.getElementById("join").style.position="absolute";document.getElementById("join").style.left="0px";document.getElementById("join").style.top="100px";intervalHandle=setInterval(animateBox, 50)}function animateBox(){currentPos+=5;document.getElementById("join").style.left=currentPos+"px";if(currentPos>900)clearInterval(intervalHandle);document.getElementById("join").style.position="",document.getElementById("join").style.left="",document.getElementById("join").style.top=""window.onload=function(){setTimeout(beginAnimate,5E3);}}
```

## Advanced

Optimization: ☐ Whitespace only ☐ Simple ☒ Advanced

[Which optimization is right for my code?](#)

Formatting: ☐ Pretty print ☐ Print input delimiter

```
// --ClosureCompiler--
// @output_file_name default.js
// @compilation_level ADVANCED_OPTIMIZATIONS
// --/ClosureCompiler--

// This is some JavaScript to run through a minifier

// create some variables
```

Compiled size: 418 bytes (428 bytes gzipped)  
Saved 58.09% off the original size (59.57% off the gzipped size)  
The code may also be accessed at [default.js](#).

Compiled Code

Warnings

Errors


POST data

```
var a=0,b;function c(){document.getElementById("join").style.position="absolute";
document.getElementById("join").style.left="0px";document.
getElementById("join").style.top="100px";b=setInterval(d,50)}function d(){a+=5;
document.getElementById("join").style.left=a+"px";if(a>900)clearInterva
l(b);document.getElementById("join").style.position="",document.getElementById(
"join").style.left="",document.getElementById("join").style.top=""})window.onload=
function(){setTimeout(c,5E3);}
```

## JavaScript Quality Checker

Jslint.com

Take code, paste into, check



The JavaScript Code Quality Tool  
Edition 2011-06-15  
[Read the instructions.](#) [Set the options.](#) [Enjoy The Good Parts.](#)

```
// This is some JavaScript to run through a code checker

// create some variables
var currentPos = 0;
var intervalHandle;

// here's where we jump the box out of the current place in the page.
function beginAnimate() {
  "use strict";
  document.getElementById("join").style.position = "absolute";
  document.getElementById("join").style.left = "0px";
  document.getElementById("join").style.top = "100px";
  intervalHandle = setInterval(animateBox,50);
}

function animateBox() {
  // set new position
  currentPos+=5;
  document.getElementById("join").style.left = currentPos + "px";
  // that's it, back to the top of the page
}
```

**Error:**

Problem at line 9 character 5: Missing 'use strict' statement.



<input type="checkbox"/> Assume console, alert, ...	<input type="checkbox"/> Tolerate bitwise operators	<input type="checkbox"/> Tolerate . and [^...] in /RegExp/
<input checked="" type="checkbox"/> Assume a browser	<input checked="" type="checkbox"/> Tolerate type confusion	<input type="checkbox"/> Tolerate misordered definitions
<input type="checkbox"/> Assume <a href="#">Node.js</a>	<input type="checkbox"/> Tolerate continue	<input type="checkbox"/> Tolerate unused parameters
<input type="checkbox"/> Assume <a href="#">Rhino</a>	<input type="checkbox"/> Tolerate debugger statements	<input checked="" type="checkbox"/> Tolerate missing 'use strict' pragma
<input type="checkbox"/> Assume a <a href="#">Yahoo Widget</a>	<input type="checkbox"/> Tolerate == and !=	<input type="checkbox"/> Tolerate inefficient subscripting
<input type="checkbox"/> Assume Windows	<input type="checkbox"/> Tolerate ES5 syntax	<input type="checkbox"/> Tolerate many var statements per function
<input type="checkbox"/> Stop on first error	<input type="checkbox"/> Tolerate eval	<input checked="" type="checkbox"/> Tolerate messy white space
<input type="checkbox"/> Safe Subset	<input type="checkbox"/> Tolerate <a href="#">unfiltered</a> for in	<input type="checkbox"/> Tolerate CSS workarounds
<input type="checkbox"/> Verify <a href="#">ADsafe</a>	<input type="checkbox"/> Tolerate uncapitalized constructors	<input type="checkbox"/> Tolerate HTML case
	<input type="checkbox"/> Tolerate dangling _ in identifiers	<input type="checkbox"/> Tolerate HTML event handlers
	<input type="checkbox"/> Tolerate ++ and --	<input type="checkbox"/> Tolerate HTML fragments

## JavaScript Libraries

- jQuery
- Closure
- Mootools
- YUI
- Dojo

## Intro to jQuery

```
<!-- loading jQuery BEFORE using it -->
<script src="jquery-1.6.1.min.js"></script>
<script src="script.js"></script>
```

```
document.getElementById("myDiv").className = "highlight";
```

```

      selector
jQuery("#myDiv").addClass("highlight");
jQuery(".someClass")
jQuery("p")           :first
jQuery("p")           :last
jQuery("a")           :contains()
jQuery("li")          :visible
jQuery("p.description")

```

```
document.getElementById("myDiv").className = "highlight";
jQuery("#myDiv").addClass("highlight");
jQuery(".someClass")
jQuery("p");
jQuery("a");
jQuery("li");

// able to drill down css selectors
jQuery("p.description");
```

```
// use jQuery - basic
//jQuery("#mainArticle").addClass("highlight");

// find all elements with a particular class
//jQuery(".tourDescription").addClass("highlight");

// find all elements with a particular tag
//jQuery("li").addClass("highlight");

// find the last li
//jQuery("li:last").addClass("highlight");

// find any paragraph that contain the word "packages"
//jQuery("p:contains('packages')").addClass("highlight");
```

## jQuery Methods

```
document.getElementById("myDiv").classname = "highlight":

jQuery("#myDiv").addClass("highlight");
                    .removeClass("highlight");
                    .toggleClass("highlight");
```

```
$("#myDiv").addClass("highlight");
$("#what to find").someAction(any params);
```

## jQuery Applications

```

// basic
document.getElementById("mainArticle").className = "highlight";

// use jQuery - basic
//jQuery("#mainArticle").addClass("highlight");

// find all elements with a particular class
//jQuery(".tourDescription").addClass("highlight");

// find all elements with a particular tag
//jQuery("li").addClass("highlight");

// find the last li
//jQuery("li:last").addClass("highlight");

// find any paragraph that contain the word "packages"
//jQuery("p:contains('packages')").addClass("highlight");

// EFFECTS

// hide all paragraphs.
//$("p").hide(4000);

//$("p").fadeOut(4000);

// EVENTS

// simple click
// avoid xpath/dom with jQuery.....
//$("#pageID").click(function() {
//    $("#pageID").text("You clicked me!");
//});

// add $(this) to refer to current element
//$("#h2").click(function() {
//    $(this).text("You clicked me!");
//});

// add effects - this makes each paragraph fade out when clicked.
//$("p").click(function() {
//    $(this).fadeOut(2000);
//});

// Page load events - instead of window.onload()
//$(document).ready(function () {
//    $("#pageID").text("The DOM is fully loaded.");
//});

// you don't have to worry about accidentally calling it multiple times.
//$(document).ready(function () {
//    $("h1").css("color", "red");
//});

```

## CONTENT DISTRIBUTION NETWORK (CDN)

- Faster to pull from Google
- Always update

- Improved bandwidth
- Improved parallel downloads ( by requesting from multiple domains )

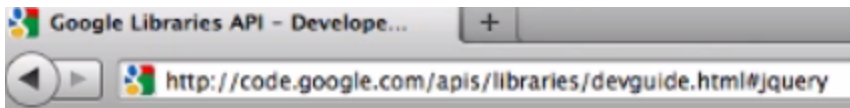
```

<div>
  <p>Regular content, etc.</p>
</div>
<script src="http://ajax.googleapis.com/ajax/
libs/jquery/1.6.1/jquery.min.js"></script>
<script src="myscript.js">
</script>
</body>
</html>

```

## Caching benefits

- Multiple websites using the CDN copy of jQuery, users won't have to load when changing sites...improved performance for client
- Don't use on internal/intranet sites where the internet may be interrupted without affecting the site



## HTTP and HTTPS PAGES

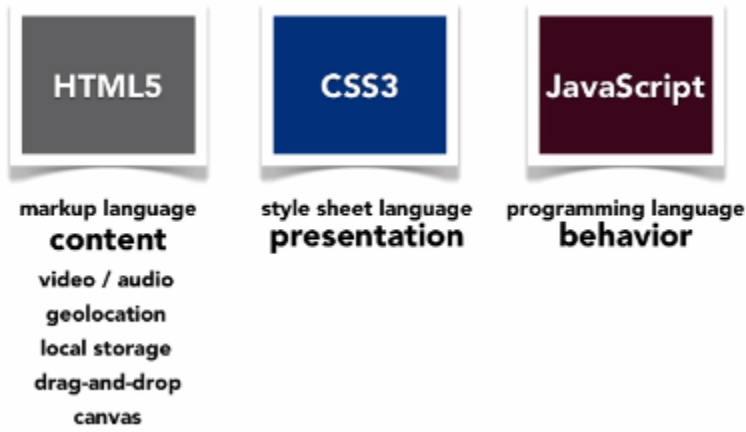
- Use whatever protocol the page is using by removing the protocol in the URL

```

<script src="//ajax.googleapis.com/ajax/libs/
jquery/1.6.1/jquery.min.js"></script>
<script src="myscript.js">
</script>

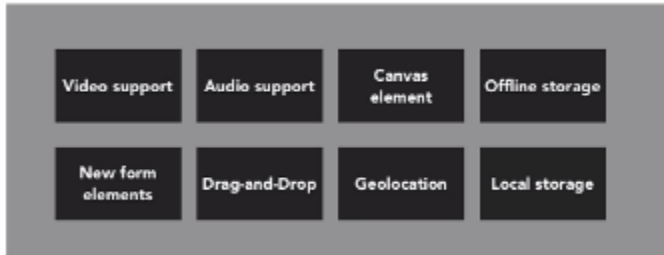
```

# HTML5 and JavaScript



## Main Features

- Varies by Browser
- Caniuse.com



## JavaScript Additions

```
var a = document.getElementById("mainTitle");  
var b = document.getElementsByTagName("li");  
  
var c = document.getElementsByClassName("myclass");  
var d = document.getElementsByClassName("first second");
```

## HTML5 VIDEO

```
<video id="myVideo" controls preload="auto">  
  <source src="videofile.mp4" type="video/mp4" />  
  <source src="videofile.webm" type="video/webm" />  
  <source src="videofile.ogv" type="video/webm" />  
  <!-- if video is not supported -->  
  <p>Your browser does not support HTML5 video.</p>  
</video>  
  
    "play" "pause"  
.play();          myVideo.addEventListener("ended", function() {  
.pause();          // code to execute when the video finishes  
.currentTime = 0;  }, false);
```

# HTML5 Storage

## HTML5 STORAGE

---

```
// set
localStorage["username"] = name;

// get
var name = localStorage["username"];
```

**Offline Storage    Web SQL    IndexedDB**

HTML5: Local Storage and Offline Applications in Depth

# Web Workers

## WEB WORKERS

---

```
var worker = new Worker("anotherjavascriptfile.js");

// get ready to receive messages from the worker
worker.onmessage = function(e) {
  console.log("The worker called me!");
};

// send messages to the worker
worker.postMessage("firstFunction");
```

# Feature Detection

```
if (document.getElementsByClassName ) {
  // it exists, we can use it
  // ...
} else {
  // it doesn't exist on this browser
}
```

## FEATURE DETECTION

---

```
if ( document.getElementsByClassName ) {
  // it exists, we can use it
  // ...
} else {
  // it doesn't exist on this browser
}
```

*[Modernizer.com](http://Modernizer.com)*

## Strict Mode

- Forces good JavaScript syntax
- Don't mix and match Strict Mode files

## JavaScript to Avoid

### Document.WRITE

- Can only use it when page is initially loading

### Browser Sniffing

- Detect features, not browsers

```
if (navigator.userAgent.indexOf('Netscape') ...
```

or

```
if (navigator.appName == 'Microsoft Internet Explorer') ...
```

(etc.)

### Eval

- Security vulnerabilities executing code that is stored in variables
- Code can be injected

### Pseudo-Protocols

- Mixing HTML and JS...bad practice
- If Javascript is turned off, won't run
- Return false; keeps link from working and allow js to handle functionality
- JS should ADD functionality

<p>Inside your HTML, you may find:

```
<a href="javascript:someFunction()">this</a>
```

</p>

<p>A preferable way is

```
<a href="nojavascript.html"
  onclick="someFunction(); return false;">this</a>
```

</p>

## Introduction to Regular Expressions

- Create the Regular Expression
- Then Apply to something else

- Looks for 'hello' to be matched against

```
var myRE = /hello/;

//or
var myRE = new RegExp("hello");
var myString = 'Does thes sentence have the work hello in it?';
if ( myRE.test(myString) ) {
    alert("Yes");
}
```

## Creating Patterns

```
var myRE = /^hello/; // ^ at the start
```

```
    /hello$/; // $ at the end
```

```
    /hel+o/; // + once or more
    "helo", "hello", "helllllllo"
```

```
    /hel*o/; // * zero or more
    "hea", "helo", "hello", "helllllllo"
```

```
    /hel?o/; // ? zero or one
    "heo", "helo" ✓
    "hello", "helllllllo"
```

```
/hello|goodbye/; // either|or
```

```
    /he..o/; // . any character
```

```
    /\wello/; // \w alphanumeric or _
```

```
    /\bhello/; // \b word boundary
```

```
    /[crnld]ope/; // [...] range of chars
    "cope", "rope", "nope", "lope", "dope" ✓
    "zope" ✗
```

## More Complex Patterns

- Email checking RegEx Examples

```
/^[0-9]{5}(:-[0-9]{4})?$/
```

```
/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/
```

## AJAX

- Asynchronous JavaScript And XML
- Ajax == JavaScript
- After web page is loaded in a browser, AJAX can call back to the server and update the page
- Create the request
- Handle the server response



- 2 different tasks so browser doesn't hang up waiting on a response

```
// Prepare the Request
var myRequest;
//feature check!
if (window.XMLHttpRequest){ //firefox, safari
    myRequest = new XMLHttpRequest();
} else if (window.ActiveXObject){ //ie
    myRequest = new ActiveXObject("Microsoft.XMLHTTP");
}

// Prepare to accept the Response
myRequest.onreadystatechange = function(){
    console.log("We were called!");
}

// THEN configure and send
myRequest.open("GET", "http://mysite.com/somedata.php", true);
myRequest.send(null);
```

## Introduction to Prototypes

- Built in language type
- Similar to 'classes'

### CREATING OBJECTS

---

```
// create an object
var playerFred = { name: "Fred", score: 10000, rank: 1 };
// add a new property
playerFred.gameType = "MMORPG";
// add a method
playerFred.logScore = function() {
    console.log(this.score);
};
// call the method
playerFred.logScore();

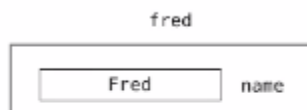
// create another object
var playerBob = { name: "Bob", highscore: 50, level: "b" };
```

### FORMALIZING OBJECTS WITH CONSTRUCTORS

---

```
function Player(n) {
    this.name = n;
}
```

```
var fred = new Player("Fred");
```



1. Create constructor function, add a passed parameter

## 2. Then call function and pass the value

```
function Player(n) {  
  // don't use specific data within the constructor  
  // use a parameter to do get the assignment value  
  //this.name = "fred";  
  this.name = n;  
}  
  
// calls the constructor function  
// var fred = new Player(); <-- no  
var fred = new Player("fred");  
var fred = new Player("jeff");
```

## Another Example

```
// Simple prototype example  
  
// creating object properties  
function Player(n,s,r) {  
  this.name = n;  
  this.score = s;  
  this.rank = r;  
}  
  
// creating object method1  
Player.prototype.logInfo = function() {  
  console.log("I am:" , this.name);  
}  
  
// creating object method2  
Player.prototype.promote = function() {  
  this.rank++;  
  console.log("My new rank is: " , this.rank);  
}  
  
var fred = new Player("Fred",10000,5);  
fred.logInfo();  
fred.promote();  
  
var bob = new Player("Bob",50,1);  
bob.logInfo();  
bob.promote();  
  
var jane = new Player("Jane",50000,10);  
jane.logInfo();  
jane.promote();
```

## Example: Countdown

Start Countdown

0:00

```

// two global variables
var secondsRemaining;
var intervalHandle;

function resetPage() {
    document.getElementById("inputArea").style.display = "block";
}

function tick() {
    // grab the h1
    var timeDisplay = document.getElementById("time");

    // turn seconds into mm:ss
    var min = Math.floor(secondsRemaining / 60);
    var sec = secondsRemaining - (min * 60);

    // add a leading zero (as a string value) if seconds less than 10
    if (sec < 10) {
        sec = "0" + sec;
    }
    // concatenate with colon
    var message = min + ":" + sec;
    // now change the display
    timeDisplay.innerHTML = message;

    // stop if down to zero
    if (secondsRemaining === 0) {
        alert("Done!");
        clearInterval(intervalHandle);
        resetPage();
    }
    // subtract from seconds remaining
    secondsRemaining--;
}

function startCountdown() {
    // get contents of the "minutes" text box
    var minutes = document.getElementById("minutes").value;
    // check if not a number
    if (isNaN(minutes)) {
        alert("Please enter a number!");
        return;
    }
    // how many seconds?
    secondsRemaining = minutes * 60;
    // every second, call the "tick" function
    intervalHandle = setInterval(tick, 1000);
    // hide the form
    document.getElementById("inputArea").style.display = "none";
}

// as soon as the page is loaded...
window.onload = function () {
    // create input text box and give it an id of "minutes"
    var inputMinutes = document.createElement("input");
    inputMinutes.setAttribute("id", "minutes");
    inputMinutes.setAttribute("type", "text");
    // create a button
    var startButton = document.createElement("input");
    startButton.setAttribute("type", "button");
    startButton.setAttribute("value", "Start Countdown");
    startButton.onclick = function () {
        startCountdown();
    }
}

```

## Example: Resize

- Resizes a page based on window.onresize and adjustStyle()

```
function adjustStyle() {
    var width = 0;
    // get the width.. more cross-browser issues
    if (window.innerHeight) {
        width = window.innerWidth;
    } else if (document.documentElement && document.documentElement.clientHeight) {
        width = document.documentElement.clientWidth;
    } else if (document.body) {
        width = document.body.clientWidth;
    }
    // now we should have it
    if (width < 600) {
        document.getElementById("myCSS").setAttribute("href", "_css/narrow.css");
    } else {
        document.getElementById("myCSS").setAttribute("href", "_css/main.css");
    }
}

// now call it when the window is resized.
window.onresize = function () {
    adjustStyle();
};
```

## CSS to hide images on small screen

```
img {
    display: none;
}
```

## Example: Accordion

- Html with #accordion css id

```
<div id="accordion">
  <h3><a href="#">Customer notifications</a></h3>
  <div>
```

```
    <p>When you book a tour with Explore California, you should receive two
notifications via email. The first will be a <strong>tour confirmation</strong>, which
states that your tour is booked, gives you the dates of your tour, and lists all
amenities included in your package. The second notification should arrive two weeks
prior to the start of your tour. This will be a <strong>reminder notification</strong>
and will contain your tour dates and current tour conditions, if applicable. <em>If you
do not receive a confirmation within 24 hours, or the reminder notification two weeks
out, contact us immediately</em>. We'll make sure there are no problems in the system
and confirm your tour.</p>
```

```
  </div>
  <h3><a href="#">Tour vouchers</a></h3>
  <div>
```

```
    <p>Some tour packages include tour vouchers. These tour vouchers allow you to
participate in optional activities during a tour and are usually scheduled for downtime
or as an optional choice to replace the day's featured activity. The vouchers are only
good during the tour and have no cash value, and cannot be redeemed if the tour is not
taken. The tour vouchers are negotiated with 3rd party vendors. Although Explore
California monitors these vendors closely, we cannot guarantee that scheduled
activities will take place.</p>
```

```
  </div>
  <h3><a href="#">Trip planning</a></h3>
  <div>
```

```
    <p>After registration, you will be sent a PDF trip planning document specific
to your tour. In the Trip Planner we offer packing advice, places of interest along the
tour route, a historical and environmental overview of the tour, a list of any required
equipment for the tour that is <em>not</em> provided by Explore California, and
additional resources for researching the surrounding area and points of interest
included in your tour. Additional information about specific tours can be found in our
FAQ section.</p>
```

```
  </div>
  <h3><a href="#">Tour checklist</a></h3>
  <div>
```

```
    <p>As you prepare for your tour, we want to make sure that you have
everything you need to fully enjoy your time in California. Having everything in place
when you arrive makes it easy to sit back and enjoy all that your tour has to offer.
With that in mind, we've prepared a small checklist to help you make sure you're ready
to go!</p>
```

```
    <ul>
      <li>Have you arranged for your mail/paper deliver?</li>
      <li>Are friends/family aware of your itinerary?</li>
    </ul>
  </div>
</div>
```

```
...
...
...
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.13/jquery-ui.min.js"></script>
<script src="script.js"></script>
</body>
</html>
```

```
window.onload = function () {  
    $("#accordion").accordion();  
};
```

## Tour vouchers

## Trip planning

After registration, you will be sent a PDF trip planning document specific to your tour.

In the Trip Planner we offer packing advice, places of interest along the tour route, a historical and environmental overview of the tour, a list of any required equipment for the tour that is *not* provided by Explore California, and additional resources for researching the surrounding area and points of interest included in your tour.

Additional information about specific tours can be found in our FAQ section.

## Tour checklist

---

## Sitelist

<http://developer.mozilla.org/en/JavaScript>

<http://dev.opera.com>

<http://jquery.com>

<http://developer.yahoo.com/javascript/>

<http://developer.yahoo.com/performance/>