

Advanced Techniques in JavaScript and jQuery

Kevin Murray
murmeister@hotmail.com



Outline

- **Traditional JavaScript functions**
 - Positional parameters
 - Handling default values
 - Object parameters

Traditional JavaScript Functions

- `function myFunction(text) { ... }`
- `myFunction = function(text) { ... }`
- `var myFunction = function(text) { ... }`
- `var myObject = { myFunction: function(text) { ... } }`
- `myObject = { myFunction: function(text) { ... } }`

Starting Simple – Draw Version 1

- **HTML5 Canvas Draw Function**
- **Recap of current draw function**
 - Contained in a global namespace
 - Doesn't accept parameters
 - Verifies the canvas context is accessible

Extending Draw – Two Parameters

- **Add some parameters**
 - Left Position
 - Top Position
- **Can call the function with different left and top values to draw a square at any location**
- **Legacy code is broken**

Two More Parameters

- **Add some more parameters**
 - Width
 - Height
- **Position and size are now definable from the calling code**
- **Canvas object may need to be sized to accommodate the values passed by the calling code**
- **Legacy code no longer works**

Six Parameters

- **Add even more parameters**
 - stroke
 - fill
- **The stroke color and fill color are now passed as parameters**
- **The Draw function requires all parameters in the proper order and with proper values**
- **Legacy code is still broken**

Fixing Legacy Code

- **Legacy code isn't passing values for the latest parameters**
 - Unexpected results
 - Errors
 - Support calls

Stubs

- **Function Stubs**

- Rename original function
- Create new function stub
- Legacy code calls stub
- New code calls new function name
- Confusing interface for new team members

Default Values

- **Provide defaults within the function when parameters are missing**
 - Useful when one or two parameters are optional
 - Optional parameters should be at the end of the parameter list
 - If more than one parameter is optional, certain assumptions must be made if only one is supplied

Default Values

- **Provide defaults for remaining parameters**
 - All parameters are optional
 - Order of parameters is still important
 - May have to pass “undefined” as a position holder

Changing to an Object Parameter

- Use one object parameter instead of multiple native types
- Objects are just named value pairs
- Properties of an object are optional by nature
- The order of object properties does not matter
- Object parameters are passed by reference

Changing to an Object Parameter

- Honor contract with legacy code
- Function logic gets more complicated when honoring previous contracts
- Object parameters simplify future expansion

Changing to an Object Parameter

- Default values as an object
- Accessible to calling code
- Can be set to simplify repetitive function calls
- Different results from legacy code – without changes
- Default values are permanently changed

Summary

- **Traditional JavaScript functions**
 - Positional parameters
 - Stub functions
 - Handling default values
 - Object parameters