

Module Pattern



Agenda

- **Module Pattern Pros & Cons**
- **Module Pattern Structure**
- **Using the Module Pattern**

Agenda

- **Module Pattern Pros & Cons**
- **Module Pattern Structure**
- **Using the Module Pattern**

The Module Pattern

■ Pros:



- “Modularize” code into re-useable objects
- Variables/functions taken out of global namespace
- Expose only public members while hiding private members

■ Cons:



- Functions may be duplicated across objects in memory when not using singleton
- Not easy to extend
- Some complain about debugging

Agenda

- **Module Pattern Pros & Cons**
- **Module Pattern Structure**
- **Using the Module Pattern**

Module Structure Overview

```
var Calculator = function() {  
    //private variables  
    //private functions  
  
    return {  
        //public members  
    };  
};
```

Module Pattern Structure

```
var Calculator = function(eq) {  
    //private member  
    var eqCtl = document.getElementById(eq);  
  
    return {  
        //expose public member  
        add: function(x,y) {  
            var val = x + y;  
            eqCtl.innerHTML = val;  
        }  
    };  
};
```

```
var calculator = new Calculator('eq');  
calculator.add(2,2);
```

Agenda

- **Module Pattern Pros & Cons**
- **Module Pattern Structure**
- **Using the Module Pattern**

Summary

- **Module pattern provides encapsulation of variables and functions**
- **Provides a way to add visibility (public versus private) to members**
- **Each object instance creates new copies of functions in memory**
- **Extending objects can be difficult since no prototyping is used**

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**