

Prototype Pattern



Agenda

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

Agenda

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

The Prototype Pattern

■ Pros:



- Leverage JavaScript's built-in features
- “Modularize” code into re-useable objects
- Variables/functions taken out of global namespace
- Functions loaded into memory once
- Possible to “override” functions through prototyping

■ Cons:



- “this” can be tricky
- Constructor separate from prototype definition

Agenda

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

Prototype Pattern Structure

```
var Calculator = function(eq) {  
    this.eqCtl = document.getElementById(eq);  
};
```

```
Calculator.prototype = {  
    add: function (x, y) {  
        var val = x + y;  
        this.eqCtl.innerHTML = val;  
    }  
}
```

```
var calc = new Calculator('eqCtl');  
calc.add(2,2);
```

Agenda

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

Summary

- **Prototype pattern leverages intrinsic JavaScript functionality**
- **Comprised of a constructor and a prototype**
- **Provides extension capabilities**

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



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