

Structuring JavaScript Code in HTML5 Applications

Dan Wahlin



Contact Info



Blog

<http://weblogs.asp.net/dwahlin>



Twitter

@DanWahlin

Why do we Need JavaScript Patterns?



Agenda

- **Function Spaghetti Code**
- **Closures to the Rescue**
- **Defining Variables**

Agenda

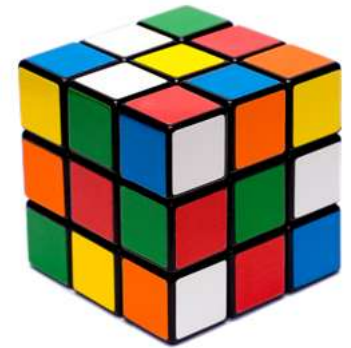
- **Function Spaghetti Code**
- **Closures to the Rescue**
- **Defining Variables**

Function Spaghetti Code



Problems with Function Spaghetti Code

- **Variables/functions added into global scope**
- **Potential for duplicate function names**
- **Not modular**
- **Not easy to maintain**
- **No sense of a “container”**



Agenda

- **Function Spaghetti Code**
- **Closures to the Rescue**
- **Defining Variables**

What is a Closure?

“...an inner function always has access to the vars and parameters of its outer function, even after the outer function has returned.”

~ Douglas Crockford

Non-Closure Example

```
function myNonClosure() {  
    var date = new Date();  
    return date.getMilliseconds();  
}
```

Variable lost after
function returns

Closure Example

```
function myClosure() {  
    var date = new Date();  
  
    //nested function  
    return function () {  
        return date.getMilliseconds();  
    };  
}
```

Variable stays around
even after function
returns

Agenda

- **Function Spaghetti Code**
- **Closures to the Rescue**
- **Defining Variables**

Defining Variables

- **Standard way to define variables:**

```
var firstName = 'James';  
var myDiv = null;
```

- **Alternative way to define variables:**

```
var firstName = 'James',  
    myDiv = null;
```

Summary

- **Function spaghetti code is bad (real spaghetti is good though!)**
- **Closures provide encapsulation**
- **Different techniques can be used to define variables**

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



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