**很好玩的东东，lazy和wrapper……**

文字没有整理完，先看代码哈 :P

[[复制]](http://bbs.51js.com/viewthread.php?tid=77486&highlight=###)

**Code:**

Function.VOID = function(){};  
Function.prototype.$lazy = function(){  
    if(this.\_\_unlazyed\_\_) return this;  
  
    var $me = this;  
    var \_cache = [];      
      
    var ret = function(){  
        var \_pointer = this;  
        var \_args = arguments;  
  
        \_cache.push(function(){  
            return $me.apply(\_pointer, \_args);  
        });  
  
        return arguments.callee;  
    }  
  
    ret.$commit =  function(){  
        var ret = null;  
        for(var i = 0; i < \_cache.length; i++){  
            ret = \_cache[i]();  
        }  
        \_cache.length = 0;  
        return ret;  
    }  
  
    ret.\_\_unlazyed\_\_ = $me;  
      
    return ret;  
}  
  
var i = 0;  
var Foo = function(a,b){  
    return i++;  
}.$lazy();  
  
var i = Foo.call().call().call().$commit();  
//alert(i);  
  
Foo();  
Foo();  
Foo();  
Foo();  
Foo();  
var j = Foo.$commit();  
  
//alert(j);  
  
//为function添加一个lazy的wrapper  
Function.prototype.$wrap = function(){  
    var $me = this;  
    var wrapperList = [];  
  
    var $ret = function(){  
        var \_args = arguments;  
        var \_callList = [];  
        var \_argsStack = [];  
        var \_cl = $me.$lazy().apply(this, \_args);  
          
        var ret = {};  
  
        ret.prevent = function(){  
            \_cl = Function.VOID.$lazy().apply(this);  
        }  
        ret.resolve = function(){  
            \_cl = $me.$lazy().apply(this, \_args);  
        }  
        ret.argsStack = \_argsStack;  
  
        for(var i = 0; i < wrapperList.length; i++){  
            var wrapper = wrapperList[i];  
            (function(wrapper,i){  
                ret[wrapper.keyword] = function(){  
                    \_callList.push(wrapper.fn.$lazy().apply(this, arguments));  
                    \_argsStack.push(arguments);              
                    return ret;  
                }  
            })(wrapper,i);  
        }  
          
        ret.$commit = function(){  
            for(var i = 0; i < \_callList.length; i++){      
                ret.\_\_index\_\_ = i;  
                \_callList[i].$commit();  
            }  
            \_cl.$commit();  
  
            \_callList.length = 0;  
            \_argsStack.length = 0;  
        }  
          
        ret.getPrevArgs = function(){  
            return ret.\_\_index\_\_ ? \_argsStack[ret.\_\_index\_\_ - 1] : \_args;  
        }  
        return ret;  
    }  
  
    $ret.addWrapper = function(keyword, fn){  
        wrapperList.push({keyword:keyword, fn:fn});  
    }  
    return $ret;  
}  
  
Foo3 = function(i){  
    alert(i);  
}.$wrap();  
  
Foo3.addWrapper("when", function(){  
    for(var i = 0; i < arguments.length; i++)   
        if (!arguments[i]){  
            this.prevent();  
            break;  
        }  
});  
  
Foo3.addWrapper("is", function(){  
    var args = this.getPrevArgs();  
    this.resolve();  
    for(var i = 0; i < arguments.length; i++)  
        if(arguments[i] != args[i]){  
            this.prevent();  
            break;  
        }  
});  
  
Foo3(10).when(false).is(false).$commit();  
  
Function.prototype.$gammar = function(){  
    var ret = this.$wrap();  
  
    ret.addWrapper("when", function(){  
        for(var i = 0; i < arguments.length; i++)   
            if (!arguments[i]){  
                this.prevent();  
                break;  
            }  
    });  
  
    ret.addWrapper("is", function(){  
        var args = this.getPrevArgs();  
        this.resolve();  
        for(var i = 0; i < arguments.length; i++)  
            if(arguments[i] != args[i]){  
                this.prevent();  
                break;  
            }  
    });  
  
    return ret;  
}  
  
var Foo2 = function(i){  
    alert(i);  
}.$gammar();  
  
var a = 10;  
var b = 20;  
Foo2(20).when(a,b).is(10,20).$commit();  
  
//Foo2(10).when(true).$commit();

  
   提示：您可以先修改部分代码再运行