

# JavaScript进阶

## ---JS中的IIFE模式



河北师范大学软件学院  
Software College of Hebei Normal University

# 内容提纲

---

- **什么是IIFE以及其使用方式**
- **通过IIFE来解决的问题（JS缺陷）**
- **IIFE实际应用案例**



## 什么是IIFE (发音: iffy)

- IIFE英文全称: Immediately-Invoked Function Expression即立即执行的函数表达式

<pre>function max(x,y){     return x&gt;y?x:y; } max(2,3); 3</pre>	<pre>(function max(x,y){     return x&gt;y?x:y; })(2,3)); 3</pre>
--	---

- IIFE的作用 (建立函数作用域, 解决ES5作用域缺陷所带来的问题, 如: 变量污染、变量共享等问题)

# IIFE的写法

- 使用小括号的写法（最常见的两种）

`(function foo( x,y){ ... }(2,3));` //2,3为传递的参数

`(function foo(x,y){ ... })(2,3);`

- 与运算符结合的写法（**执行函数、进行运算**）

`var i = function( ){ return 10; }();` //i为10

`true && function( ){ ... }();`

`~function(arg1,arg2){ ... }(x,y);` //x,y为传递参数 位运算非操作符

`!function( ){ ... }();` //思考 `!function(){return 2; }();` 与 `!function(){return 0; }();`

```
> !function(){  
    return 2;  
}();
```

```
<< false
```

```
> !function(){  
    return 0;  
}();
```

```
<< true
```

# 内容提纲

---

- 什么是IIFE以及其使用方式
- 通过IIFE来解决的问题 (JS缺陷)
- IIFE实际应用案例



# 通过IIFE来解决JS缺陷

## •通过IIFE对作用域的改变（限制变量生命周期）

- JS（ES5）中没有块级作用域，容易造成js文件内或文件间的同名变量互相污染
- 我们往往会通过IIFE引入一个新的作用域来限制变量的作用域，来避免变量污染

```
var userId = 12;  
document.onclick = function () {  
    console.log("userId = ",userId);  
};
```

//一长串代码后，忘记了上边定义的userId

```
var a=2,b=3;  
if(a<b){  
    var userId = 34;//重复定义，变量污染  
}
```

参见实例：  
demo07\_1  
前半部分

同一文件内  
的变量污染



# 通过IIFE来解决JS缺陷

## • 通过IIFE对作用域的改变（限制变量生命周期）

- JS（ES5）中没有块级作用域，js文件内和文件间的同名变量容易互相污染
- 我们往往会通过IIFE引入一个新的作用域来限制变量的生命周期

```
//(function () { // IIFE开始
    var x = 10;
    document.onclick = function () {
        console.log("x = ",x);
    };
//})(); // IIFE结束
```

文件1中的 通过立即执行表达式来避免变量污染  
代码

**思考：**如果不用立即执行表达式，而是直接写函数，然后再调用，是否可以实现同等效果

```
//(function () { // IIFE开始
    var x = 20;
//})(); // IIFE结束
```

文件2中的  
代码，污染了文件1中的x

参见实例：  
demo07\_1后半部分  
demo07\_2  
index07\_1\_2.html  
不同文件之间的变量污染

# 非期望的变量共享问题及解决办法

## • 通过IIFE对变量存储的改变（避免变量共享错误）

- 当程序运行到变量所在作用域时，变量被创建，JS（ES5）没有块作用域，变量可能会共享
- 如下例：在函数作用域中创建的变量i只有一个，出现了变量i共享问题，可通过IIFE解决

```
function f(){  
    var getNumFuncs = []; // 函数数组  
    for(var i=0; i<10; i++){  
        (function (j) {  
            getNumFuncs[j] = function(){return j;};  
        })(i);  
    }  
    return getNumFuncs;  
}  
var tmp = f();  
tmp[3](); // 输出为3, tmp[0]()...tmp[9]()都为期望的结果
```

参见实例demo08  
和index08  
变量共享及解决

查看Scope窗体中  
getNumFuncs中每  
一个函数的内部属性  
[[Scopes]]中的闭包  
中的变量，看是否存  
在共享问题





# 内容提纲

---

- 什么是IIFE以及其使用方式
- 通过IIFE来解决的问题（JS缺陷）
- **IIFE实际应用案例**



## IIFE实际引用案例（页面导航问题）

- 避免闭包中非期望的变量共享问题

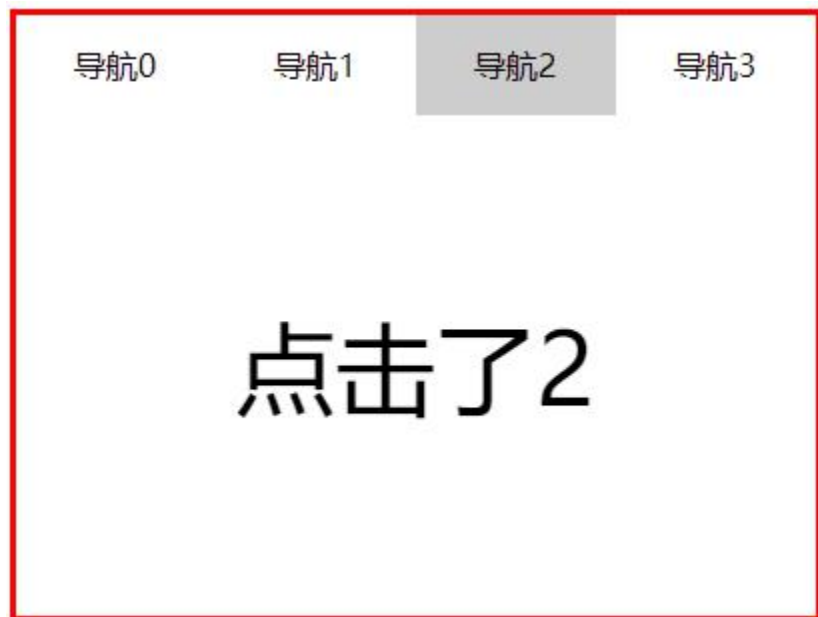


```
for(var i=0;i<tabs.length;i++){  
    //!function(i){        //IIFE start  
        tabs[i].onclick=function(){  
            for (var j = 0; j < tabs.length; j++) {  
                tabs[j].className = '';  
            }  
            this.className = 'active';  
            contents.innerHTML = "点击了"+i;  
        }  
    //}(i);                //IIFE end  
}
```

tab的length为4，由于变量共享在同一个作用域下，所以变量*i*只有一个，并最终*i*为4，所以点击任何标签，都输出“点击了4”

## IIFE实际引用案例（页面导航问题）

- 避免闭包中非期望的变量共享问题，解决方式 IIFE



```
for(var i=0;i<tabs.length;i++){
    !function(i){    //IIFE start
        tabs[i].onclick=function(){
            for (var j = 0; j < tabs.length; j++) {
                tabs[j].className = '';
            }
            this.className = 'active';
            contents.innerHTML = "点击了"+i;
        }
    }(i);            //IIFE end
}
```

tab的length为4，立即执行了4次函数，有4个函数作用域，所以变量*i*生成了4次，所以点击时能正常输出1到4

## IIFE实际引用案例（定时器案例）

- 避免闭包中非期望的变量共享问题，解决方式 IIFE

```
for (var i = 0; i < 3; i++) {  
    (function(j) { // j = i  
        setTimeout(function() {  
            console.log(new Date, j);  
        }, 1000*i);  
    })(i);  
}
```

undefined

Mon Sep 04 2017 16:51:26 GMT+0800 (中国标准时间) 0

Mon Sep 04 2017 16:51:27 GMT+0800 (中国标准时间) 1

Mon Sep 04 2017 16:51:28 GMT+0800 (中国标准时间) 2

# 总结

---

- 什么是IIFE以及其使用方式
- 通过IIFE来解决的问题（JS缺陷）
- IIFE实际应用案例



The background of the slide is decorated with various abstract shapes in shades of green and yellow. These shapes, which include circles, ovals, and teardrop-like forms, are scattered across the top and right sides of the slide, creating a modern and organic feel.

# Thank You!



河北师范大学软件学院  
Software College of Hebei Normal University

# 作业

---

- codefordream网站上JavaScript基础
- 雪梨上完成任务（要求有截图，体现完成的项目，用户名，完成的程度）
- 复习本章内容及练习

## 闭包的常见形式（函数作为参数）

- 综合实例（闭包、高阶函数、静态词法作用域、IIFE）

```
var max = 10;  
var fn = function (x) {  
    if(x > max){  
        console.log(x);  
    }  
};  
(function (f) {  
    var max = 100;  
    f(15);  
})(fn);
```

左侧实例输出什么？  
使用断点调试查看代码的运行状况