

理解JavaScript中的闭包

Frank Zhang

Tech Leader @ EMC

博客

<http://blog.csdn.net/zijinshi>

个人主页

<http://delphi.zijinshi.org>

邮件

franczx@hotmail.com

内容

- 变量的作用域
- 闭包的定义
- 闭包的例子
- 闭包的应用
- 性能考虑

变量的作用域

在JavaScript中，局部变量的作用域是由它定义的函数决定，嵌套函数可以访问它的外部作用域的变量。

代码： ex1

```
1 function init() {  
2     var name = "Mozilla"; // name is a local variable created by init  
3     function displayName() { // displayName() is the inner function, a closure  
4         alert(name); // use variable declared in the parent function  
5     }  
6     displayName();  
7 }  
8 init();
```

运行结果

Mozilla

闭包的定义

Closures are functions that refer to independent (free) variables. In other words, the function defined in the closure 'remembers' the environment in which it was created.

-- From Mozilla Developer Network

闭包是涉及独立变量的函数。换句话说，在闭包中定义的函数会记住它创建的环境。

代码： ex2

```
14 function makeFunc() {  
15     var name = "Mozilla";  
16     function displayName() {  
17         alert(name);  
18     }  
19     return displayName;  
20 }  
21 var myFunc = makeFunc();  
22 myFunc();
```

运行结果

Mozilla

闭包的例子

代码: ex3

```
28 function makeAdder(x) {  
29   return function(y) {  
30     return x + y;  
31   };  
32 }  
33  
34 var add5 = makeAdder(5);  
35 var add10 = makeAdder(10);  
36  
37 console.log(add5(2));  
38 console.log(add10(2));
```

运行结果

7

12

function(y) {
 return 5 + y;
};

function(y) {
 return 10 + y;
};

使用闭包

当希望某些对象只能被一个方法使用时，我们可以使用闭包。

代码： ex4

JavaScript

```
56 function makeSizer(size) {
57   return function() {
58     document.body.style.fontSize = size + 'px';
59   };
60 }
61
62 var size12 = makeSizer(12);
63 var size14 = makeSizer(14);
64 var size16 = makeSizer(16);
65
66 document.getElementById('size-12').onclick = size12;
67 document.getElementById('size-14').onclick = size14;
68 document.getElementById('size-16').onclick = size16;
```

```
42 body {
43   font-family: Helvetica, Arial, sans-serif;
44   font-size: 12px;
45 }
46
47 h1 {
48   font-size: 1.5em;
49 }
50
51 h2 {
52   font-size: 1.2em;
53 }
```

```
71 <p>Some paragraph text</p>
72 <h1>some heading 1 text</h1>
73 <h2>some heading 2 text</h2>
74
75 <a href="#" id="size-12">12</a>
76 <a href="#" id="size-14">14</a>
77 <a href="#" id="size-16">16</a>
```

使用闭包 模拟私有方法

代码: ex5_1

```
81 var counter = (function() {
82     var privateCounter = 0;
83     function changeBy(val) {
84         privateCounter += val;
85     }
86     return {
87         increment: function() {
88             changeBy(1);
89         },
90         decrement: function() {
91             changeBy(-1);
92         },
93         value: function() {
94             return privateCounter;
95         }
96     };
97 })();
98
99 console.log(counter.value()); // logs 0
100 counter.increment();
101 counter.increment();
102 console.log(counter.value()); // logs 2
103 counter.decrement();
104 console.log(counter.value()); // logs 1
```

代码: ex5_2

```
108 var makeCounter = function() {
109     var privateCounter = 0;
110     function changeBy(val) {
111         privateCounter += val;
112     }
113     return {
114         increment: function() {
115             changeBy(1);
116         },
117         decrement: function() {
118             changeBy(-1);
119         },
120         value: function() {
121             return privateCounter;
122         }
123     };
124 };
125
126 var counter1 = makeCounter();
127 var counter2 = makeCounter();
128 alert(counter1.value()); /* Alerts 0 */
129 counter1.increment();
130 counter1.increment();
131 alert(counter1.value()); /* Alerts 2 */
132 counter1.decrement();
133 alert(counter1.value()); /* Alerts 1 */
134 alert(counter2.value()); /* Alerts 0 */
```

使用闭包

常见错误：在循环中创建闭包

代码： ex6

JavaScript

```
139 function showHelp(help) {
140     document.getElementById('help').innerHTML = help;
141 }
142
143 function setupHelp() {
144     var helpText = [
145         {id: 'email', 'help': 'Your e-mail address'},
146         {id: 'name', 'help': 'Your full name'},
147         {id: 'age', 'help': 'Your age (you must be over 16)'}
148     ];
149
150     for (var i = 0; i < helpText.length; i++) {
151         var item = helpText[i];
152         document.getElementById(item.id).onfocus = function() {
153             showHelp(item.help);
154         }
155     }
156 }
157
158 setupHelp();
```

```
function makeHelpCallback(help) {
    return function() {
        showHelp(help);
    };
}
```

```
document.getElementById(item.id).onfocus = makeHelpCallback(item.help);
}
```

HTML

```
166 <p id="help">Helpful notes will appear here</p>
167 <p>E-mail: <input type="text" id="email" name="email"></p>
168 <p>Name: <input type="text" id="name" name="name"></p>
169 <p>Age: <input type="text" id="age" name="age"></p>
```


性能考虑

代码: ex7_1

```
167 function MyObject(name, message) {  
168     this.name = name.toString();  
169     this.message = message.toString();  
170     this.getName = function() {  
171         return this.name;  
172     };  
173  
174     this.getMessage = function() {  
175         return this.message;  
176     };  
177 }
```

ex7_2

```
180 function MyObject(name, message) {  
181     this.name = name.toString();  
182     this.message = message.toString();  
183 }  
184 MyObject.prototype = {  
185     getName: function() {  
186         return this.name;  
187     },  
188     getMessage: function() {  
189         return this.message;  
190     }  
191 };
```

ex7_3

```
194 function MyObject(name, message) {  
195     this.name = name.toString();  
196     this.message = message.toString();  
197 }  
198 MyObject.prototype.getName = function() {  
199     return this.name;  
200 };  
201 MyObject.prototype.getMessage = function() {  
202     return this.message;  
203 };
```

ex7_4

```
206 function MyObject(name, message) {  
207     this.name = name.toString();  
208     this.message = message.toString();  
209 }  
210 (function() {  
211     this.getName = function() {  
212         return this.name;  
213     };  
214     this.getMessage = function() {  
215         return this.message;  
216     };  
217 }).call(MyObject.prototype);
```

相关资料

- 参考资料 <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>
- 幻灯片 http://delphi.zijinshi.org/downloads/courses/1/Understanding_Closures_in_JavaScript.pdf
- 源代码 <http://delphi.zijinshi.org/downloads/courses/1/code.txt>

THANKS

本课程由 Frank Zhang 提供

<http://delphi.zijinshi.org>

CSDN网站：www.csdn.net

企业服务：<http://ems.csdn.net/>

人才服务：<http://job.csdn.net/>

CTO俱乐部：<http://cto.csdn.net/>

高校俱乐部：<http://student.csdn.net/>

程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>

项目外包：<http://www.csto.com/>

CSDN博客：<http://blog.csdn.net/>

CSDN论坛：<http://bbs.csdn.net/>

CSDN下载：<http://download.csdn.net/>