基础实验 2 - JavaScript 基础

软件 21
赵雷彧
2012013285
Zly.george@163.com

一、基础练习一

返回如下函数的返回值：

```
var func=
{
    getNum: function() {return this.num;},
    num:1
};
(function(){return typeof arguments[0]();})(func.getNum);
```

返回值：undefined

原因：

考察关于 this 的含义： In JavaScript, as in most object-oriented programming languages, this is a special keyword that is used within methods to refer to the object on which a method is being invoked. (jQuery Fundamentals (Chapter 2), by Rebecca Murphey) 同时缺省值为 global(或 window)，故在此处由于调用 getNum 是通过 arguments[0]实现的,故其 this 指针指向 arguments. 又由于 arguments 对象无 num 成员，故输出为 undefined.

更进一步，如果代码如下

```
var func=
{
    getNum: function() {return this.num;},
    num:1
};
(function(a){return typeof a();})(func.getNum);
```

则由于 this 指针缺省，指向 global(浏览器中为 Window)。而当代码如此时

```
var func=
{
    getNum: function() {return this.num;},
    num:1
};
(function(){return typeof func.getNum();})();
```

this 指向 func，得到输出为 num。

以上代码均在 chrome 控制台得到验证。

二、基础练习二

如下 console.log 出的是什么：

```
var x=0;
function foo()
{
    x++;
    this.x=x;
    return foo;
}
var bar = new new foo;
console.log(bar.x);
```

输出值：undefined

原因：

在此实例中，foo 用作 constructor 以期待生成新的 object，然而在 foo 的末尾出现了 return foo;当 return 一个 object 或 function 时，将替代 new 的返回值。故 new foo 与 foo 等价，故此语句中 new 没有意义，bar==foo。bar.x 当然不存在。

进一步，`var bar = new new new … new foo;` 的结果也应当完全一致。

而作为 constructor 的正确用法，例如

```
var x=0;
function foo()
{
    x++;
    this.x=x;
    //return 123;
}
var bar = new foo; //显然，两个new会出现语法错误
console.log(bar.x);
```

将会得到 bar 作为一个由 foo 新建的 object，其属性为{x:1}。故输出为 1。

以上代码均在 chrome 控制台得到验证。

三、基础练习三

如下 alert 的结果是什么：

```
function bar()
{
    return foo;
    foo=10;
    function foo(){}
    var foo='11';
}
alert(typeof bar());
```

输出值：function

原因：

由于 function 定义被脚本解释器预编译，即使解释到某一步时未出现相应函数的定义也可以像已经定义过一般使用。相应的，变量却不存在预编译，故另外两个变量赋值行为均未被执行，最终得到的 foo 为 function

以上代码均在 chrome 控制台得到验证。

四、基础练习四

如下 alert 的结果是什么：

```
var x=3;
var foo=
{
    x:2,
    baz:
    {
        x:1,
        bar: function(){return this.x;}
    }
}
var go=foo.baz.bar;

alert(go());
alert(foo.baz.bar());
```

输出值：第一个 alert: 3； 第二个 alert: 1。

原因：

同基础练习一，不多做赘述。go 使得 this 指针指向 global，而 foo.baz.bar 使 this 指向 foo.baz。

以上代码均在 chrome 控制台得到验证。

五、基础练习五

如下 alert 的结果是什么：

```
function aaa()
{
    return
    {
        test:1
    };
}
alert(typeof aaa());
```

输出值：undefined

原因：

Return 的返回值必须跟于其后，否则当做 return;看。自然为 undefined。

更进一步，如果按照如下两种方式书写代码，便会返回期望的值(即 object)：

```
function aaa()
{
    return {
        test:1
    };
}
alert(typeof aaa());
```

```
function aaa()
{
    return {test:1};
}
alert(typeof aaa());
```

以上代码均在 chrome 控制台得到验证。

六、进阶练习一

通过预设 A1~H2 与数字 0~15 的映射数组(代码中的 HASH, ARCHASH)方便互相转换，同时对比赛赛程的每一场比赛都保存一个数组表示第 i 支队伍出现于此的概率，而初始位显然为只有其自己出现概率为 1 其余都为 0 的数组。

而后通过函数按赛程模拟每一场比赛逐步算出出现于冠军的概率数组,在其中输出待查询队伍的概率。

代码详见 src/forecast.js，代码后有附简单测试数据。

七、进阶练习二

由于比对函数具有重载的特征，故单独提取出比对函数进行伪重载，利用 typeof 确定其类型，如果为 number 或 string 分别处理，而为 object 时利用 in 操作符对三个属性一一检查存在性并作比对。为方便特判按姓名查找时只输出第一个人，函数返回值分三种而非两种。

代码详见 src/search.js，代码后有附简单测试数据。

八、UNTRUSTED 吐槽向攻略

关卡	攻略	代码链接
1	删掉那一坨创造障碍的代码即可	https://gist.github.com/57f2efebe3d15a432b02
2	第一个插入点将 map 重定位成一个没有实质意义的类，第二个插入点将 map 恢复	https://gist.github.com/e6d9ed045bbb4c9046d9
3	更改填充物使其满足下限同时还能不阻碍通向终点的路	https://gist.github.com/c56d9b2640631e559798
4	新加一个出口啊…我去居然可以这样..	https://gist.github.com/e6d08e9ddea6c3b3b575
5	setSquareColor 把 mine 标出来	https://gist.github.com/545acb9d53ad25d1c41e

6	加一个小管道把炸弹困在里面	https://gist.github.com/d8ec359c66785f273a38
7	电话函数周期变色即可	https://gist.github.com/3e93b54cc482aaa2bd6a
8	打电话不停重新生成森林从里面找路	https://gist.github.com/e0ed7888baa023babff5
9	打电话在合适的时候改变船的航向	https://gist.github.com/08329c39681e66ebc3df
10	改变炸弹行为，上下两行一直右飞，中间一行有机会就上飞。空出中间一行。	https://gist.github.com/79ae1ec57b13787c0890
11	机器人的移动规则：能向下则向下，否则向右	https://gist.github.com/dbcf9d2e14fbafea2eca
12	手动在数组里给机器人制定前进路线	https://gist.github.com/069a48ef168f34a41558
13	DFS 了一发..(早写这个代码前两关还费什么劲儿啊)	https://gist.github.com/85be0621badd5d3e0357
14	蛋疼的设定在于钥匙不能重复拥有，所以下方两个小房间显然是无效的，利用没有某种颜色(红蓝)钥匙时也可通过绿门的设定可以节省一把这种颜色的钥匙。 从而节省一把蓝色钥匙用于最后。	https://gist.github.com/a929142091ccfbe023eb
15	神坑!! 居然利用 throw 机制来让人不死... 要不是我放弃治疗随便填东西在里面... 我觉得这游戏我快不能玩了...	https://gist.github.com/d8b3a88187b710f10be5
16	居然不坑.... 画激光的时候把颜色也画出来然后打电话改颜色就好	https://gist.github.com/5d3b95ca0a2110b74233
17	选定某个离出口近的传送门，把所有的传送门都定向到他	https://gist.github.com/373b3e9662c31a02a31d
18	跳的时候设置一个每 45ms 把自己向上抬一下的飞天函数就好(打个电话都下不来了)	https://gist.github.com/91671755d3d1b51998c6
19	(什么东西啊这关我什么都没干随便按了几个键看看是怎么回事代码都没看就过了= =)	https://gist.github.com/997f28ea6481a7175ef2
20	定义一个弹幕攻击 biubiubiu 干掉 boss，电话被收缴了，所以 overRide 左键吧	https://gist.github.com/73cb376b2a0b46282655
21	menu+里的其他代码看了半天，最后只有各种调戏能调戏的代码，后来找到了 object::exit 那玩意.. 改改代码把最终关 flag 判定关了居然就过了	https://gist.github.com/f08d9e369d731043385f
22	完结撒花！Bilibili （ °- °)つロ 乾杯！	N/A