

课程基础实验二文档

卢晨辉

2014013438

Luch14@126.com

一、 基础练习

1. 基础练习 1

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

运行结果：undefined

原因分析：在上面代码段中用强制运算符运算“函数直接量声明”这个表达式，并返回一个函数自身的引用，然后通过函数调用运算符“()”来操作这个函数引用。对参数 func.getNum 进行操作，由于 func.getNum 返回的结果是 this.num，而当前作用域中并不存在 num 变量，因此结果是 undefined。

2. 基础练习 2

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

运行结果：undefined

原因分析：第一次执行 new foo 的过程中，由于 foo 的返回值仍然是 foo，因此 foo 内的其他内容均被舍弃，获得的是 foo，而在执行第二次 new

操作的时候，情况与第一次是相似的。由于关于 x 的内容被舍弃，因此 bar,x 是未定义的，结果为 undefined。

3. 基础练习 3

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

运行结果：function

原因分析：虽然 return foo 语句在 function(){}之前，但是在执行之前其实已经做过预编译处理，创建出所需要的活动对象，因此在执行的过程中可以直接返回 foo，由于 foo 是一个 function 类型的，因此在最终执行的结果中，得到的结果是 function 类型的

4. 基础练习 4

```
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());
```

运行结果：3 1

原因分析：在执行 alert(go())语句时，this 指针指向的是 window，在 window

下 this.x 的值为 3，因此 alert 的结果为 3。在执行 alert(foo.baz.bar())时，this 指针指向的对象是 foo.bar，此时 x 的值为 1，因此 alert 的结果为 1。

5. 基础实验 5

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

运行结果：undefined

原因分析：由于 javascript 会将每行结尾自动视作语句的结尾，因此第 2 行 return 语句相当于 return;语句，从而可以得到返回的是一个并不存在的内容。因此 aaa()的结果是 undefined 类型的，alert 出的结果是 undefined。

二、进阶练习

1. Forecast 问题

算法简要说明：实现该问题主要使用了 3 个函数。Probability 函数主要实现的功能是计算两只球队对战时，根据战力值所得到的其中一只球队晋级的概率，即 $p = a / (a + b)$ 。calculate 函数主要实现的是计算某一轮比赛发生后，各队晋级的概率，例如有四只球队 A : 20, B: 80, C, 40, D: 60. 那么第一轮 calculate 过后分别对应数组[0.2, 0.8, 0, 0]和[0, 0, 0.4, 0.6]，第二轮 calculate 过后，A 的值为 $0.2 * 0.4 * (20 / (20+40)) + 0.2 * 0.6 * (20 / (20+60))$ ，以此类推可以计算出每个球队对应的晋级概率。故在 forecast 函数中进行的过程是根据对阵情形进行各轮的 calculate 计算，

最终获得十六强的所有夺冠概率 (forecast 函数的两个参数分别为包含各队队名和战力值的 Object 和国家名字的字符串), 最终返回查询国家对应的概率。

2. Search 问题

算法简要说明：实现该问题使用了 1 个函数：search 函数。Search 函数实现的函数重载功能主要是通过对第 2 个参数的类型进行判断来实现的。当判断第 2 个参数的类型为数字时，执行以下内容：遍历存有学生信息的数组，寻找是否有年龄与该数字相同的个体，若有则加入结果数组，最后判定结果数组是否为空，若为空，返回 false，否则返回结果数组。当判断第 2 个参数类型为字符串时，遍历存有学生信息的数组，当找到一个 name 与该字符串相同的个体时，返回该个体，结束运算，若遍历完仍未发现符合要求的，返回 false。当判断第 2 个参数类型为 object 时，遍历寻找是否有包含符合 object 中所有内容的个体，若有，则加入结果数组，后判定结果数组是否为空，若为空，返回 false，否则返回结果数组。