

JavaScript 基础练习

WEB 前端实训 基础作业 2

2014013412 杨景

18569095589@163.com

目录

1.基础练习	1
1.1 基础练习 1-Arguments,作用域	2
1.2 基础练习 2-动态属性,构造函数返回值	2
1.3 基础练习 3-typeof,预编译	2
1.4 基础练习 4-this 指针	2
1.5 基础练习 5-语句	2
2.进阶练习	2
2.1 进阶练习 1-forecast 方法	2
2.2 进阶练习 2-search 方法	3
3.BONUS	2
3.1 BONUS1-diff 方法	2
3.2 BONUS2-排序算法	3

1 基础练习

1.1 基础练习 1-Arguments,作用域

【问题】如下的执行函数返回值是什么？并解释原因。

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

(function(){
  return typeof arguments[0]();
})(func.getNum);
```

【回答】该函数返回 undefined。原因为：this 是包含它的函数作为方法被调用时所属的对象，在下面的函数调用 func.getNum 的方法时，this 指的就是下面函数中的对象，而下面这个对象没有 num 这个值，故为 undefined。

1.2 基础练习 2-动态属性,构造函数返回值

【问题】如下 console.log 输出什么？并解释原因。

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

【回答】console.log 输出 undefined。原因为：当一个函数的返回值是引用类型（数组，对象或者函数）的数据，那么这个函数作为构造函数用 new 运算符执行构造时，运算的结果将被它的返回值取代，这时候，构造函数体内的 this 值丢失了，取而代之的是被返回的对象，因而第二次返回 foo 就是当前的对象 bar，有 this 值丢失，故 bar.x 为 undefined。

1.3 基础练习 3-typeof,预编译

【问题】如下 alert 结果是什么？并解释原因。

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

【回答】alert 结果为 function。原因为：函数声明时会在脚本执行之前进行预编译，而变量赋值在预编译处理阶段只是分配一个空间，不做初始化，在执行时再赋值初始化。因而当题中函数进行预编译时，会将 foo 作为一个 function 进行处理，从而执行 typeof 时返回 function。

1.4 基础练习 4-this 指针

【问题】如下 alert 的结果分别是什么？并解释原因。

```

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());

```

【回答】 alert 分别返回 3 和 1。原因为：this 是包含它的函数作为方法被调用时所属的对象，go 为全局变量，故 this 指向 window，即为 x = 3，所以输出 3。而 foo.baz.bar() 为简单的对象中函数的调用，会沿着作用域链往上找，故输出 1。

1.5 基础练习 5-语句

【问题】 alert 输出结果是什么？并解释原因。

```

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

```

【回答】alert 输出 undefined。因为：

2 进阶练习

2.1 进阶练习 1-forecast 方法

代码详见源码区，forecast.js。

实现思路：首先定义赢球概率，即

$\text{某队的赢球概率} = \text{某队实力} / \text{两队实力之和}$

在求出某队是最后冠军的概率之前，我分别求出所有球队第一次赢球，第二次赢球，以及第三次赢球的概率，从而冠军即是第四次赢球的球队，即可方便求出。

2.2 进阶练习 2-search 方法

代码详见源码区，search.js。

实现思路：根据第二个参数的类型进行判定。

- 第一种情况，第二个参数为数字，遍历学生数组，若符合，则进入输出数组中。判定输出数组长度，为 0，则输出 false，反之输出这个数组即可。

- 第二种情况，第二个参数为字符串，同第一种情况的算法。

- 第三种情况，第二中参数为对象，遍历学生数组，然后遍历第二个对象中的属性，若两者共有的所有属性相同，则添加到输出数组中。判定输出数组长度，为 0，则输出 false，反之输出这个数组即可。

3 BONUS

3.1 BONUS1-diff 方法

代码详见源码区，diff.js。

实现思路：遍历第二个学生参数，然后遍历第一个学生参数，比较时只比价 name 属性，不同则添加到输出数组中，输出这个数组即可。

3.2 BONUS2-排序算法

代码详见源码区，sort.js。

实现思路：分别实现冒泡排序，插入排序和选择排序。分别参照这些算法的实现思想即可完成。