

task19

问题

1.基础类型有哪些？复杂类型有哪些？有什么特征？

基本类型值：（字符串、数值、布尔值、null和undefined）：指的是保存在栈内存中的简单数据段；

引用类型值：（对象、数组、函数、正则）：指的是那些保存在堆内存中的对象，变量中保存的实际上只是一个指针，这个指针执行内存中的另一个位置，由该位置保存对象。

2.如下代码的输出？为什么？

```
var obj1 = {a:1, b:2};
var obj2 = {a:1, b:2};
console.log(obj1 == obj2);//false
console.log(obj1 = obj2);//Object {a: 1, b: 2}
console.log(obj1 == obj2);//true
```

//一开始2个obj指向2个不同的堆内存位置，因此虽然内容相同，但位置不同；但第二个console使得两者指向了同一位置，所以第三次是true

代码

1.写一个函数getIntv，获取从当前时间到指定日期的间隔时间

```
var str = getIntv("2016-12-29");
function getIntv(str){
  var space = (Date.parse(str)-Date.now())/1000;
  var day = Math.floor(space/60/60/24);
  var hour = Math.floor((space - day*60*60*24)/60/60);
  var min = Math.floor((space - day*60*60*24 - hour*60*60)/60);
  var sec = Math.floor(space - day*60*60*24 - hour*60*60 - min*60);
  console.log ("距" + str + "还有 " + day + " 天 " + hour + " 小时 " + min + " 分 " + sec + " 秒");
}
```

2.把数字日期改成中文日期

```

var str = getChsDate('2015-02-08');
console.log(str); // 二零一五年一月八日
function getChsDate(num){
    var time = new Date(num);
    var year = time.getFullYear()+'';
    var month = time.getMonth()+1+'';
    var date = time.getDate()+'';
    var chinese =['零','一','二','三','四','五','六','七','八','九','十','十一','十二','十三','十四','十五','十六','十七','十八','十九','二十','二十一','二十二','二十三','二十四','二十五','二十六','二十七','二十八','二十九','三十','三十一'];
    var n = '';
    for(var i =0;i<year.length;i++){
        n = n+chinese[year[i]];
    }
    return n+'年'+chinese[month]+'月'+chinese[date]+'日';
}

```

3.写一个函数获取n天前的日期

```

var lastWeek = getLastNDays(7);
function getLastNDays(num) {
    var a = new Date(Date.now() - num*1000*3600*24);
    var b = [a.getFullYear(),a.getMonth()+1,a.getDate()];
    console.log(b.join("-"));
}

```

4.完善如下代码，用于获取执行时间如：

```

var Runtime = (function(){
    var start,end,get;
    var obj = {
        start: function(){
            start = Date.now();    //code here ... , 当前时间
            return start;
        },
        end: function(){
            end = Date.now(); //code here ... 结束时间
            return end;
        },
        get: function(){
            get = end - start; //code here ... 获取执行时间
            return get/1000;
        }
    };
    return obj;
})();

```

```
Runtime.start();
for (var i=0;i<1000;i++) {
    console.log('Hello World');
}
//todo somethint
Runtime.end();
console.log('说出1000个Hello World用了'+ Runtime.get()+'秒' );
```

5.楼梯有20级，每次走1级或是2级，从底走到顶一共有多少种走法？用代码（递归）实现

```
function get(n){
    if(n===0||n===1){    //0级和1级都是1种
        return 1;
    }else{
        return get(n-1)+get(n-2);
    }
}
get(20);
//10946
```

6.写一个json对象深拷贝的方法，json对象可以多层嵌套，值可以是字符串、数字、布尔、json对象中的任意项（PS:尝试另外一种方法

```
var obj2 = JSON.parse( JSON.stringify(obj1) )
```

//深拷贝

```
var Chinese = {nation:'中国',birthPlaces: ['北京','上海','香港']};
var Doctor = {career:'医生'};
```

```
function deepCopy(p, c) {
    var c = c || {};
    for (var i in p) {
        if (typeof p[i] === 'object') {
            c[i] = (p[i].constructor === Array) ? [] : {}; //三元运算符?:
            deepCopy(p[i], c[i]);
        } else {
            c[i] = p[i];
        }
    }
    return c;
}
```

```

}

var Doctor = deepCopy(Chinese);
Doctor.career = '医生';

Doctor.birthPlaces=['北京','上海','香港','厦门'];
console.log(Doctor.birthPlaces); //北京, 上海, 香港, 厦门
console.log(Chinese.birthPlaces); //北京, 上海, 香港
console.log(Doctor);

////////////////////////////////////

JSON.stringify(obj1) )
function deepCopy(oldObj){
    var newObj=oldObj;
    if (oldObj&&typeof oldObj==="object") {
        newObj=    Object.prototype.toString.call(oldObj)=== "[object Array]
"? []: {};
        for(var i in oldObj){
            newObj[i]=deepCopy(oldObj[i]);
        }
    }
    return newObj;
}

function deepCopy(oldObj){
    var newObj=JSON.stringify(oldObj);
    return JSON.parse(newObj);
}

```



Leanote
Upgrade Account