js面试经典题

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

return的用法，return，break，continue后面换行的话，会默认的加；

function foo1()

{

 return {

 bar: "hello"

 };

}

function foo2()

{

 return   =====>换行会自动结束

 {

 bar: "hello"

 };

}

foo1：{bar:’hello’} foo2:underfine

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

循环输出

for(var i = 0; i < 5; i++) {

 (function(i) {

 setTimeout(function() {

 console.log(i);

 }, 1000);

 })(i)

}

会输出 0 1 2 3 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

绑定事件的方法

IE采用

attachEvent("onclick",function(){}) detachEvent("onclick",function(){})

其他采用

addEventListener("click",function,false)

removeEventListener("click",function,false)

阻止时间冒泡

IE：eve.returnValue=false

其他：eve.preventDefault();

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(function(){

 var a = b = 3;

})();

a: underfine  b:3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

console.log(NaN === NaN); //false

NaN不等于自己

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

0.1+0.2=0.300000000000000000

要想0.1+0.2=0.3

function add(num1, num2){

 let r1, r2, m;

 r1 = (''+num1).split('.')[1].length;

 r2 = (''+num2).split('.')[1].length;

 m = Math.pow(10,Math.max(r1,r2));

 return (num1 \* m + num2 \* m) / m;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1+ + ‘1’ ==>2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

递归

(function fn(n){

return n<=1?1:n\*fn(n-1);

})(5);

或者

function fac(num){

return num<=1?1:num\*arguments.callee(num-1);

 }

fac(5);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

数组去重处理

function sort(arr){

var result={};

var newArr=[];

for(var i=0;i<arr.length;i++){

if(!result[arr[i]]){

newArr.push(arr[i]);

result[arr[i]]=1;

}

}

return newArr;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

判断质数

function isPrime1(n){

if(n<=3){return true}

else{

for(var i=2;i<Math.sqrt(n);i++){

if(n%i==0){return false;}

}

return true;

}

};

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

查看字符串中出现最多字符以及个数

var str="helloworld";

方法一：用hash

for(var i=0,hash={};i<str.length;i++){

if(hash[str[i]]){

hash[str[i]]++

}else{

hash[str[i]]=1;

}

}

console.dir(hash);

方法二：用正则

var arr=str.split("")

.sort()

.join("")

.match(/([a-z])\1\*/g)

.sort(function(a,b){

return b.length-a.length; })

console.log("出现最多的是: "+arr[0][0]

+"共"+arr[0].length+"次");

var hash={};

arr.forEach(function(val){

hash[val[0]]=val.length;

});

console.dir(hash);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*