|  |
| --- |
| Northern Lights (Tian Jin) |
| Javascript Basic Quiz |
| For QA |

|  |
| --- |
| Trainer Wang, Bai Kang (Northern Lights)  2-4-2018 |

本测验为检验QA团队自学javascript成果，共29题均为问答题，每题1分共29分，按60%的通过率答对18题即可通过，请仔细阅读题目，答案自在其中，祝大家顺利通过测验。

1. 请写段代码调用方法console.log()输出"Hello World"（不包括双引号）

代码：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 分析下段代码输出结果

var arr = [2, 3, 4, 5, 6];

var sum = 0;

for (var i = 1; i < arr.length; i++) {

sum += arr[i]

}

console.log(sum);

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 以下代码运行的输出结果

var a = b = 10;

(function () {

var a = b = 20

})();

console.log(b);

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 以下代码运行后的输出结果

var a = [1, 2, 3];

console.log(a.join());

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 在 JavaScript 中，'1555' + 3的运行结果

结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 请写段代码调用方法alert()在页面上弹出"Hello World"（不包括双引号）

代码：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 以下代码运行后页面弹出的结果

var a = 888;

++a;

alert(a++);

弹出的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 请写出下列的表达式的真假值，回答真或假，true或 false均可

!(3 <= 1)

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(4 >= 4) && (5 <= 2)

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

("a" == "a") && ("c" != "d")

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(2 < 3) || (3 < 2)

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"12" > "9"

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 请写出下列的表达式的真假值，回答真或假，true或 false均可

parseInt(12.5) == parseFloat(12.5)

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number('') == parseFloat('')

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

isNaN('abc') == NaN

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

typeof NaN === 'number'

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

null == undefined

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

undefined == false

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

undefined === undefined

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NaN == NaN

值：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读下面代码，计算k的运行结果

var i = 0, j = 0;

for (; i < 10, j < 6; i++, j++) {

k = i + j;

}

k的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读以下代码，计算y的运行结果

var x = 1; function fn(n){n = n+1}; y = fn(x);

y 的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [1, 2, 3, 4].join('0').split('') 的执行结果

结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 以下代码运行后，计算arr和arr2的运行结果

var arr = [1, 2];

var arr2 = arr.concat();

arr2.push(arr.splice(1, 0));

arr的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

arr2的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 分析下面的代码，计算输出的结果

var arr = new Array(5);

arr[1] = 1;

arr[5] = 2;

console.log(arr.length);

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读以下代码，计算n的运行结果

var n = "Northern Lights Corporation".indexOf("Northern", 6);

n的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读以下代码，写出页面弹出的结果

var s = "abcdefg";

alert(s.substring(1, 2));

弹出的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读以下代码，写出页面弹出的结果

function fn1() {

alert(1);

}

alert(fn1());

第一次弹出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第二次弹出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 阅读一下代码，是否会弹出结果？是，请写出结果；否，请写出原因。

fn1();

var fn1 = function (a) {

alert(a);

}

是否会弹出结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

结果/原因：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 以下代码中，写出会出现什么结果

function fn1() {

var a = 0;

function fn2() {

++a;

alert(a);

}

return fn2;

}

fn1()();

var newFn = fn1();

newFn();

newFn();

第一次弹出的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第二次弹出的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第三次弹出的结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码的运行结果

var a = 100;

function fn1() {

alert(a);

var a = 10;

}

alert(fn1());

第一次弹出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第二次弹出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码运行结果

var a = 100;

function fn1() {

let a = 10;

console.log(a);

}

fn1();

console.log(a);

第一次输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第二次输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 分析下面代码执行结果，如果没有结果请写出原因

const PI = Math.PI;

PI = 23

结果：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码运行结果

class Animal {

constructor() {

this.type = 'animal';

}

says(say) {

console.log(this.type + ' says ' + say);

}

}

let animal = new Animal();

animal.says('hello');

class Cat extends Animal {

constructor() {

super();

this.type = 'cat';

}

}

let cat = new Cat();

cat.says('hello');

第一次输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

第二次输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码运行结果

let fn1 = (x, y) => {

x++;

y--;

return x + y;

}

console.log(fn1(10, 10));

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 请写出下面代码运行结果

let words = 'Hello!';

console.log(`${words} Northern Lights!`);

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 请写出下面代码运行结果

let dog = {type:'animal',age:2};

let {type,age} = dog;

console.log(type,age);

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码运行结果

function animals(type = 'cat') {

console.log(type);

}

animals();

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 写出下面代码运行结果

function animals(…types) {

console.log(types);

}

animals('dog', 'cat', 'fish');

输出：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 字符串 " northern lights " 调用trim()方法返回"northern lights"即去掉字符串前后空格。若用正则表达式实现，请写出能够查找出前后空格的正则表达式

正则表达式：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_