

# 家庭作业

## 一、选择题

1. 如何遍历下面的 my\_data 对象？

```
var my_data={a:'Ape', b:'Banana', c:'Citronella'};
```

- A. for(var key in my\_data) {}
- B. my\_data.forEach(function(item){})
- C. for(var key = 0;key<my\_data.length;key++){}
- D. 以上都不对

**D**

2. 判断一个js对象是否是Array,arr为要判断的对象，下面可行的方法是？【多选】

- A. typeof arr
- B. arr instanceof Array
- C. arr.toString()==='[object Array]'
- D. Object.prototype.toString.call(arr)==='[object Array]'

**B**

3. 下面那些会返回false

- A. 1 == true
- B. "" == false
- C. false == null
- D. null == undefined

**D**

4. +new Array(017) 这段代码输出为？

- A. 17
- B. 15
- C. NaN
- D. 报错

**C**

5. 执行以下程序，下列说法中，正确的是（）【多选】

```

var arr = new Array(3); ...①
arr[0] = 1;
arr.b = 0;
console.log(arr.length); ...②
arr.forEach(function(value){
    console.log(value); ...③
})
for(var i in arr){
    console.log(arr[i]); ...④
}

```

- A. ①中创建一个长度为3的数组
- B. ②中输出结果为4
- C. ③中输出结果为1 0
- D. ④中输出结果为0 1 2 b length

A

#### 6. 阅读一下代码

```

var name="World!";
(function(){
    var name;
    if(typeof name === 'undefined'){
        name='Jack';
        console.log('Goodbye'+name);
    }
    else{
        console.log('Hello'+name);
    }
})();

```

- A. Hello World!
- B. Goodbye Jack
- C. Hello Jeck
- D. Goodbye World!

A

#### 7. 如果以下程序的输出结果是false, 则①式可以替换为 ( )

```

const test = {
    rules: false
};
function Build() {
    this.rules = true;
    ①
}
const build = new Build();
console.log(build.rules);

```

- A. return false;
- B. return this.rules;
- C. return test;
- D. 什么都不写

## C

### 8. 阅读一下代码

```
var a="undefined";
var b="false";
var c="";
function assert(aVar){
    if(aVar)
        console.log(true);
    else
        console.log(false);
}
assert(a);
assert(b);
assert(c);
```

- A. true, true, true
- B. true, true, false
- C. false, false, true
- D. false, false, false

## B

### 9. 一下代码会输出什么?

```
var a = 4399 < 0 || typeof(4399 + '');
console.log(a);
```

- A. true
- B. string
- C. undefined
- D. false
- E. number

## B

### 10. 一下代码的输出结果是什么?

```
var aaa = 12;
function fun1(){
    console. log(aaa);
    var aaa = 20;
    console.log(aaa);
}
fun1();
```

- A. 12 20
- B. 12 12
- C. undefined 20
- D. 20 undefined

## 二、分析题

注：所有看程序写结果的题目都要求写出分析过程

1. 如何让 `(a == 1 && a == 2 && a == 3)` 的值为true?

```
let v=1;
let a={
  toString(){
    return v++;
  }
}
```

2. 看程序写结果

```
var str = "1949新中国成立了! ";
var arr = [];
for(var i = 0; parseInt(str[i]); i++){
  arr[i] = str[i];
}
console.log(arr); //["1","9","4","9"]
```

3. 看程序写结果

```
var length=10;
function fn(){
  console.log(this.length);
}
var obj={
  length:5,
  method:function(fn){
    fn();
    arguments[0]()
  }
}
obj.method(fn,1); //10 2
```

4. 看程序写结果

```
var v = 123;
function foo(){
  var v = 456;
  function inner(){
    console.log(v);
  }
  return inner;
}
result = foo()
console.log(result()) //456
```

5. 看程序写结果

```
function Father(age){
    this.age = age
}
function Son(age){
    Father.call(this);
}
Son.prototype = Father.prototype;
Father.prototype.getAge = function(){
    console.log(40);
}
Son.prototype.getAge = function(){
    console.log(18);
}
var father = new Father(40);
var son = new Son(18);
son.getAge(); //18
father.getAge(); //18
```

6. 看程序写结果

```
for(var i=0;i<5;++i){
    setTimeout(function(){
        console.log(i+ ' '); // "5"
    },100);
}
```

7. 看程序写结果

```
var obj = {};
obj.log = console.log;
obj.log.call(console,this);//window
```

8. 阅读如下代码

```
var x= 10;
var y= 20;
var z= x < y ? x++ : ++y;
// x,y,z最后的值分别是多少
//11 20 10
```

9. 读代码，写结果

```
var arr1 = [10, 20];
arr1.push(30);
arr1.push([40, 50]);
var data= arr1.pop();
console.log(data);           // 输出什么? [40,50]
```

10. 下面的代码会输出什么?

```

console.log(typeof (function(){})); //function
console.log(typeof ['标哥', '帅哥']); //object
console.log(typeof null); //object
console.log(typeof undefined); //undefined
console.log(typeof Function.prototype); //function
console.log('标哥' instanceof String); //false
console.log(new Date() instanceof Date); //true

```

11. 下面代码会输出什么

```

var obj = {
  num1: 117
}
var res = obj;
obj.child = obj = { num2: 935 };
var x = res.child.num2;
console.log(obj.child); //undefined
console.log(res.num1); //117
console.log(x); //935

```

### 三、编程题

1. 写一个函数，生成一个随机 IP 地址，一个合法的 IP 地址为 0.0.0.0 ~ 255.255.255.255。

```

function randomNum(min, max) {
  return Math.floor(Math.random() * (max - min + 1) + min);
}

function randomIP() {
  let arr = [];
  for (let i = 0; i < 4; i++) {
    arr.push(randomNum(0, 255));
  }
  return arr.join(".")
}
console.log(randomIP());

```

2. 将数字12345678转换成RMB形式。如：12,345,678

```

//RMB
function parseRMB(num) {
  let result = [];
  while (true) {
    if (num % 1000 === num) {
      result.unshift(num % 1000)
      break;
    }
    result.unshift(num % 1000)
    if (num > 1000) {
      num = ~~(num / 1000);
    }
  }
}

```

```

    }
    return result.join(",");
}
console.log(parseRMB(12345678));

```

3. 删除与某个字符相邻且相同的字符，比如 "fdaffdaaklfjklja"字符串处理之后成为"fdafdaklfjklja"

```

//删除相邻
function removeRe(str) {
    let arrStr = str.split("");
    console.log(arrStr);
    for (let i = 0; i < arrStr.length; i++) {
        if (arrStr[i] === arrStr[i - 1]) {
            delete arrStr[i]
        }
    }
    return arrStr.join("");
}

console.log(removeRe("fdaffdaaklfjklja"));

```

4. 一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。

```

function test(num){

    //返回true或false,如果是回文数就返回true,如果不是就返回false
    //参数num就是传进来需要判断的数
    //补齐方法里面的代码体
    num = (num + "").split("");
    if (num.reverse() === num) {
        return true
    } else {
        return false
    }
}

```

5. 请设计并写出一个函数，反转一个数组。该函数具有一个参数用以传入数组，并返回一个新的数组，不允许用到reverse函数。

```

//反转
function reverseArr(arr) {
    let newArr = [];
    arr.forEach((i) => {
        newArr.unshift(i);
    });
    return newArr;
}
console.log(reverseArr([1, 2, 3, 4, 5]));

```

6. 请设计一个 f1flatten 的函数，实现“拍平”一个多维数组为一维。示例如下

```

var testArr1 = [[0, 1], [2, 3], [4, 5]];
var testArr2 = [0, [1, [2, [3, [4, [5]]]]]];
flatten(testArr1) // [0, 1, 2, 3, 4, 5]
flatten(testArr2) // [0, 1, 2, 3, 4, 5]

function flatten(arr){
    //在这里补充代码
    //返回结果
    if (Array.isArray(arr)) {
        let newArr = [];
        function flattenInner(arr1) {
            if (Array.isArray(arr1)) {
                arr1.forEach((i) => {
                    return flattenInner(i);
                })
            } else {
                newArr.push(arr1);
                return arr1;
            }
        }
        flattenInner(arr);
        return newArr;
    } else {
        console.error("请输入数组");
    }
}

```

7. 请设计并写出一个函数，请将一个数组中的数字挑选出来，并返回所有数字的和，如果数组中无数字则返回 undefined。该函数具有一个参数用以传入字符串。

```

function numSum(arr) {
    let numArr = [];
    let result = 0;
    arr.forEach((i) => {
        if (typeof i === "number" && !Number.isNaN(i)) {
            numArr.push(i);
        }
    })
    if (numArr.length !== 0) {
        numArr.forEach((i) => {
            result += i;
        })
    } else {
        result = undefined;
    }
    return result;
}

```

8. 设计一个函数 isInteger(x) 来判断 x 是否是整数。

```

function isInteger(x) {
    return x === ~~(x);
}

```



9. 以下是一数组，请将数组按要求拆分。

```
//原数组
var arr =["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o"]
//结果数组
var newArr = [
  ["a","b"],
  ["c","d","e"],
  ["f","g"],
  ["h","i","j"],
  ["k","l"],
  ["m","n","o"]
]
let newArr2=[];
newArr2.push(arr.slice(0, 2));
  newArr2.push(arr.slice(2, 5));
  newArr2.push(arr.slice(5, 7));
  newArr2.push(arr.slice(7, 10));
  newArr2.push(arr.slice(10, 12));
  newArr2.push(arr.slice(12, 15));
console.log(newArr2)
```

10. 编程翻转单词"I am a student !"变为"! student a am i"

```
// 翻转单词
function reverseWord(str) {

  return str.split(" ").reverse().join(" ");
}
console.log(reverseWord("I am a student !"));
```

11. 有一母牛，到4岁可生育，每年一头，所生均是一样的母牛，到15岁绝育，不再能生，20岁死亡，问20年后有多少头牛

```
let cows = [0];
for (let i = 0; i < 20; i++) {
  cows.forEach((i, index) => {
    cows[index] = ++cows[index];
    if (i > 4 && i < 15) {
      cows.push(0);
    }
  });
}
console.log(cows.filter(i => i < 20).length);
```

12. 随机生成一个N\*N矩阵(用二维数组模拟,整个数组内不允许有重复数字)并找出鞍点（行最大的数，列最小的数）  
例如：

```
[  
  [1,9,4],  
  [8,2,5],  
  [7,6,3]  
]
```

不存在鞍点

```
[  
  [1,2,3],  
  [4,5,6],  
  [7,8,9]  
]
```

鞍点为3

```
//审题审不明白...
```