一、选择题，一题2分，共40分

1. JavaScript是一种：

A. 服务器端语言

B. 客户端语言

C. 桌面应用语言

D. 嵌入式系统语言

答案：B

2. 在JavaScript中，全局变量声明关键字是：

A. var

B. let

C. const

D. function

答案：A

3. 下列哪个不是JavaScript的原始数据类型？

A. Number

B. String

C. Object

D. Array

答案：C

4. JavaScript中，用于控制循环的语句不包括：

A. for

B. while

C. break

D. switch

答案：D

5. 下列哪个方法用于数组元素的添加？

A. push()

B. pop()

C. shift()

D. unshift()

答案：A

6. 在JavaScript中，`null`和`undefined`的区别是什么？

A. 它们是相同的，只是写法不同

B. `null`是一个表示“没有值”的特殊值，而`undefined`表示变量已声明但未初始化

C. `null`用于数字，`undefined`用于字符串

D. `null`表示数组的结束，`undefined`表示函数的结束

答案：B

7. JavaScript中，`==`和`===`的区别是什么？

A. 没有区别，它们的行为是相同的

B. `==`是严格等于，`===`是非严格等于

C. `==`会进行类型转换，`===`不会

D. `==`用于基本类型，`===`用于对象类型

答案：C

8. 下列哪个方法用于删除数组中的最后一个元素？

A. push()

B. pop()

C. shift()

D. unshift()

答案：B

9. JavaScript中，`setTimeout`函数用于：

A. 立即执行代码

B. 在指定的毫秒数后执行代码

C. 每间隔指定的毫秒数执行代码

D. 停止执行代码

答案：B

10. 下列哪个方法用于删除数组中的特定元素？

A. splice(index, 1)

B. remove(index)

C. delete(index)

D. pop(index)

答案：A

11. JavaScript中，`arguments`对象包含什么？

A. 函数的参数

B. 函数的局部变量

C. 函数的属性

D. 函数的调用堆栈

答案：A

12. 下列哪个不是JavaScript的全局对象？

A. Math

B. Date

C. String

D. Array

答案：C

13. JavaScript中，`typeof`操作符用于：

A. 检查一个变量是否已声明

B. 检查一个变量的类型

C. 检查一个变量是否已初始化

D. 检查一个变量是否是函数

答案：B

14. 下列哪个方法用于将数组连接成字符串？

A. join()

B. concat()

C. slice()

D. splice()

答案：A

15. 在JavaScript中，如何创建一个匿名函数？

A. function()

B. function anonymous()

C. () => {}

D. new Function()

答案：C

16. 下列哪个不是JavaScript的原型链对象？

A. Object.prototype

B. Function.prototype

C. Array.prototype

D. String.prototype

答案：B

17. JavaScript中，`==`和`===`操作符的区别是什么？

A. `==`是严格等于，`===`是非严格等于

B. `==`会进行类型转换，`===`不会

C. `==`用于基本类型，`===`用于对象类型

D. `==`用于数字，`===`用于字符串

答案：B

18. JavaScript中，`Function.prototype.bind`方法的作用是什么？

A. 创建一个新函数

B. 创建一个新函数，将其`this`关键字设置为提供的值

C. 创建一个新函数，忽略前几个参数

D. 创建一个新函数，改变其执行环境

答案：B

19. 下列哪个方法用于立即执行代码？

A. setTimeout()

B. setInterval()

C. setImmediate()

D. IIFE（Immediately Invoked Function Expression）

答案：D

20.在JavaScript中，使用XMLHttpRequest对象发送请求时，哪个方法用于发送请求？

A. connect()

B. open()

C. send()

D. request()

答案：C

1. 判断题，一题2分，共20分

当然可以，这里是10个关于JavaScript的判断题：

1. 在JavaScript中，对象字面量中可以包含函数作为其属性。（ ）

答案：正确

2. JavaScript中的`var`关键字声明的变量具有块级作用域。（ ）

答案：错误（它具有函数作用域或全局作用域）

3. JavaScript中的arguments对象包含了函数调用时传入的所有参数，但它并不是一个真正的数组，因此不支持数组的方法。（ ）

答案：正确

4. 在JavaScript中，使用`==`运算符比较两个值时，会进行类型转换。（ ）

答案：正确

5.JavaScript中，`null`和`undefined`是同一个概念，只是写法不同。（ ）

答案：错误（`null`表示没有值，而`undefined`表示值未定义）

6. 在JavaScript中，`Array.prototype.push()`方法用于在数组末尾添加一个或多个元素，并返回新的长度。（ ）

答案：正确

7. `JSON.stringify()`方法可以将JavaScript对象转换为JSON字符串，但它不能序列化函数或循环引用的对象。（ ）

答案：正确

8. 在JavaScript中，`Math.floor()`方法总是向下取整，即使传入的数字是负数。（ ）

答案：正确

9. `fetch` API返回的是一个`Promise`对象，这意味着你可以使用`.then()`和`.catch()`方法来处理成功和失败的情况。（ ）

答案：正确

10.在JavaScript中，使用`delete`操作符可以删除对象的属性，但不会删除对象本身。（ ）

答案：正确

1. 代码题，一题10分，🟰30分
2. 编写一个JavaScript函数 isPalindrome，该函数接受一个整数作为参数，并检查该整数是否是回文数（即正序和倒序都一样的数）

例如：

console.log(isPalindrome(121)); // 应该输出 true

console.log(isPalindrome(123)); // 应该输出 false

function isPalindrome(num) {

const str = num.toString();

const reversedStr = str.split('').reverse().join('');

return str === reversedStr;

}

// 测试

console.log(isPalindrome(121)); // 应该输出 true

console.log(isPalindrome(123)); // 应该输出 false

1. 编写一个JavaScript函数 findMostFrequent，该函数接受一个数组作为参数，并返回出现次数最多的元素。如果多个元素出现次数相同且都是最多的，返回其中任意一个即可。

例如：

console.log(findMostFrequent([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])); // 输出4

function findMostFrequent(arr) {

const frequencyMap = {};

let maxCount = 0;

let mostFrequent;

arr.forEach(item => {

if (frequencyMap[item]) {

frequencyMap[item]++;

} else {

frequencyMap[item] = 1;

}

if (frequencyMap[item] > maxCount) {

maxCount = frequencyMap[item];

mostFrequent = item;

}

});

return mostFrequent;

}

// 测试

console.log(findMostFrequent([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])); // 应该输出 4

3.编写一个JavaScript函数 calculateDaysBetween，该函数接受两个日期字符串作为参数，并返回这两个日期之间的天数差。假设输入的日期格式为 YYYY-MM-DD。

例如；

console.log(calculateDaysBetween('2024-01-01', '2024-01-15')); // 应该输出 14

function calculateDaysBetween(dateStr1, dateStr2) {

const date1 = new Date(dateStr1);

const date2 = new Date(dateStr2);

const timeDiff = Math.abs(date2.getTime() - date1.getTime());

const dayDiff = timeDiff / (1000 \* 3600 \* 24);

return Math.floor(dayDiff);

}

// 测试

console.log(calculateDaysBetween('2023-01-01', '2023-01-15')); // 应该输出 14