

前端JS笔试面试题

前端JS笔试面试题

笔试部分

Q: 下面有关javascript内部对象的描述，正确的有？（多选）

Q: 以下哪个是错误的（单选）

Q: 写出以下程序的输出结果，为什么？

Q: 有完成以下代码，使其输出结果与注释中的一样（忽略时区信息）

Q: 以下程序输出什么，为什么？

Q: 请写出以下程序的输出结果

Q: 在 javascript 中，用于阻止默认事件的默认操作的方法是？

Q: 在标准的 JavaScript 中， Ajax 异步执行调用基于下面哪一项机制才能实现(单选)，并写出选择该项的理由

Q: 请写出可以匹配以下16进制颜色值中任意一个的正则表达式

Q: 请写出正则表达式，匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)

Q: 请输出以下程序的执行结果

Q: 请写出以下程序的执行结果

Q: 请写出以下程序的输出结果

Q: 以下程序输出什么，为什么？

Q: 请写出以下程序的执行结果，为什么？

Q: 请在下方列出var与let的区别

Q: 假设当前网址是 `http://foouser:barpassword@www.wrox.com:80/WileyCDA/?q=javascript#contents` 请写出以下属性的值：

Q: 请写一个方法来判断一个任意值是否为文本节点

Q: 请写一个方法来判断一个对象是否为可迭代对象

Q: 请使用正则写一个可以过滤文本中所有的 `script` 标签的函数

Q: 请写出以下程序的输出结果

Q: 请编写一个方法，实现给出任意一个数组，可以去重后返回

Q: 请完成 `flatten` 方法

Q: 请实现一个可兼容所有浏览器的DOM事件监听函数

Q: 请完成以下程序：

Q: 请实现 `getCommonAncestor` 方法，返回任意两个DOM节点的最近的公共祖先节点

Q: 请实现 `Person` 方法，使其可以通过如下方式被调用

Q: 请使用高阶函数分贝完成以下需求

Q: 列出JS中常见的错误类型，其中哪些是早期错误，并写一个示例程序，抛出一个错误并捕获

Q: 写出以下程序的执行结果

Q: `<script>` 标签的 `async` 属性与 `defer` 属性之间的异同

Q: 请写出以下程序的执行结果，为什么？

Q: 请写出以下程序的执行结果，为什么

Q: 请完成 `parseQuery` 方法，要求必须使用正则

Q: 请完成 `concatWithSort` 方法：

Q: 以下代码，请使用至少两种不同方法实现 `a` 与 `b` 的值互换，要求不使用临时变量，并给出解题思路

面试部分

Q: 请简述前后端分离架构的优缺点

Q: 请简述前端都有哪些安全问题，怎么解决

Q: 浏览器缓存读取规则

Q: 简述浏览器的工作流程

Q: `cookie` 和 `token` 都存放在 `header` 中，为什么不会劫持 `token`？

Q: 介绍模块化发展历程

笔试部分

Q: 下面有关javascript内部对象的描述，正确的有？（多选）

1. History 对象包含用户（在浏览器窗口中）访问过的 URL
2. Location 对象包含有关当前 URL 的信息
3. Window 对象表示浏览器中打开的窗口
4. Navigator 对象包含有关浏览器的信息

Q: 以下哪个是错误的（单选）

1. iframe是用来在网页中插入第三方页面，早期的页面使用iframe主要是用于导航栏这种很多页面都相同的部分，这样在切换页面的时候避免重复下载
2. iframe的创建比一般的DOM元素慢了1-2个数量级
3. iframe标签会阻塞页面的加载
4. iframe本质是动态语言的Include机制和利用ajax动态填充内容

Q: 写出以下程序的输出结果，为什么？

```
1 (function() {  
2  
3     var x=foo();  
4  
5     var foo=function foo() {  
6         return "foobar"  
7     };  
8     return x;  
9 })();
```

Q: 有完成以下代码，使其输出结果与注释中的一样（忽略时区信息）

```
1 var d = new Date('2018-05-09');  
2 //在下面编写代码  
3 d.setDate(40);  
4  
5 console.log(d)// Sat Jun 09 2018 xxxxx (xxxx)  
6
```

Q: 以下程序输出什么，为什么？

```
1 function Foo(){  
2     'use strict'  
3     console.log(this.location);  
4 }  
5 Foo();
```

Q: 请写出以下程序的输出结果

```
1 console.log(typeof Date.now());
```

Q: 在 javascript 中，用于阻止默认事件的默认操作的方法是？

Q: 在标准的 JavaScript 中， Ajax 异步执行调用基于下面哪一项机制才能实现(单选)，并写出选择该项的理由

1. Event和callback
2. 多线程操作
3. 多CPU核
4. Deferral和promise

Q: 请写出可以匹配以下16进制颜色值中任意一个的正则表达式

```
1 #ffbbad
2 #Fc01DF
3 #FFF
4 #ffE
```

Q: 请写出正则表达式，匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)

Q: 请输出以下程序的执行结果

```
1 console.log(Function.prototype.__proto__.__proto__ === null);
```

Q: 请写出以下程序的执行结果

```
1 var a =[1,2,3];
2 var b = a.slice(0,1);
3 b.push(4);
4 console.log(a, b)
```

Q: 请写出以下程序的输出结果

```
1 var a = 10;
2 var obj = {
3   a: 20,
4   fn: function(a){
5     this.a += 5;
6     let b = 3;
7     function fn(a) {
8       this.a += 2;
9       a += b;
10      console.log(this.a + a);
11    };
12    return fn;
13  }
14 };
15
16 var fn = obj.fn(a);
17 fn(3);
18 fn(5);
19 fn(obj.a);
```

Q: 以下程序输出什么，为什么？

```

1  var tmp = new Date();
2
3  function f() {
4      console.log(tmp);
5      if (false) {
6          var tmp = "hello world";
7          console.log(tmp);
8      }
9      console.log(tmp);
10 }
11
12 f();

```

Q: 请写出以下程序的执行结果，为什么？

```

1  a = 1;
2  console.log(window.a);
3
4  let b = 1;
5  console.log(window.b);

```

Q: 请在下方列出**var**与**let**的区别

Q: 假设当前网址是

`http://foouser:barpassword@www.wrox.com:80/wileyCDA/?`

`q=javascript#contents` 请写出以下属性的值：

- `location.hash`
- `location.search`
- `location.pathname`
- `location.origin`
- `location.username`

Q: 请写一个方法来判断一个任意值是否为文本节点

Q 请写一个方法来判断一个对象是否为可迭代对象

Q: 请使用正则写一个可以过滤文本中所有的**script**标签的函数

Q: 请写出以下程序的输出结果

```

1  let [foo, , , [[bar], , baz]] = [1,2,3,4,['a','b'], 'e', {a:1}];
2  console.log(foo);
3  console.log(bar);
4  console.log(baz);

```

Q: 请编写一个方法，实现给出任意一个数组，可以去重后返回

Q: 请完成 `flatten` 方法

```
1 function flatten(arr){
2
3 }
4 var arr = flatten([1,2,[3,4,[5,6,[7,[8]]]]]);
5 console.log(arr);
6 // 输出: [ 1, 2, 3, 4, 5, 6, 7, 8 ]
```

Q: 请实现一个可兼容所有浏览器的DOM事件监听函数

Q: 请完成以下程序:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="initial-scale=1, maximum-scale=3, minimum-scale=1,
6   user-scalable=no">
7   <title>canvas保存图片</title>
8   <script>
9     window.onload = function() {
10       draw();
11       var dlButton = document.getElementById("downloadImageBtn");
12       bindButtonEvent(dlButton, "click", saveAsLocalImage);
13     };
14     function draw(){
15       var canvas = document.getElementById("thecanvas");
16       var ctx = canvas.getContext("2d");
17       // 绘制一个边长为100的正方形，并填充为透明度50%的黑色,位置随意
18
19       ctx.fillText("canvas图片下载", 50, 50);
20     }
21
22     function bindButtonEvent(element, type, handler)
23     {
24       if(element.addEventListener) {
25         element.addEventListener(type, handler, false);
26       } else {
27         element.attachEvent('on'+type, handler);
28       }
29     }
30
31     function saveAsLocalImage () {
32       var myCanvas = document.getElementById("thecanvas");
33       // 完成该方法，使其可以将canvas内容下载到本地
34     }
35   </script>
36 </head>
37 <body bgcolor="#E6E6FA">
38 <div>
```

```
39     <canvas width=200 height=200 id="thecanvas"></canvas>
40     <button id="downloadImageBtn">下载为图片</button>
41 </div>
42 </body>
43 </html>
```

Q: 请实现`getCommonAncestor`方法，返回任意两个DOM节点的最近的公共祖先节点

```
1 function getCommonAncestor (nodeA, nodeB) {
2
3 }
```

Q: 请实现`Person`方法，使其可以通过如下方式被调用

```
1 var xiaoming = Person('xiaoming', 23);
2 xiaoming.name().age();
3 // 输出:
4 // xiaoming
5 // 23
```

Q 请使用高阶函数分贝完成以下需求

```
1 var arr = [
2   {
3     id:1,
4     name: '张三',
5     score: 80
6   },
7   {
8     id:2,
9     name: '李四',
10    score: 60
11  },
12  {
13    id:3,
14    name: '王五',
15    score: 75
16  },
17  {
18    id:4,
19    name: '陈六',
20    score: 80
21  },
22  {
23    id:5,
24    name: '鬼脚七',
25    score: 80
26  },
27  ];
```

1. 返回一个字符串，字符串中是所有score为80的人的ID和名称，如 1-张三，用逗号分隔。
2. 计算所有人得分的总和

Q: 列出JS中常见的错误类型，其中哪些是早期错误，并写一个示例程序，抛出一个错误并捕获

Q: 写出以下程序的执行结果

```
1  async function async1() {
2      console.log('async1 start');
3      await async2();
4      console.log('async1 end');
5  }
6  async function async2() {
7      console.log('async2');
8  }
9  console.log('script start');
10 setTimeout(function() {
11     console.log('setTimeout');
12 }, 0)
13 async1();
14 new Promise(function(resolve) {
15     console.log('promise1');
16     resolve();
17 }).then(function() {
18     console.log('promise2');
19 });
20 console.log('script end');
```

Q: <script>标签的async属性与defer属性之间的异同

Q: 请写出以下程序的执行结果，为什么？

```
1  var a = ['a1', 'a2']
2  var b = a
3  [0,1].slice(1)
4  console.log(b)
```

Q: 请写出以下程序的执行结果，为什么


```

1  { // 外层块
2    let x = "outer";
3    { // 内层块
4      console.log(x);
5      var refX1 = function () { return x };
6      console.log(refX1());
7      const x = "inner";
8      console.log(x);
9      var refX2 = function () { return x };
10     console.log(refX2());
11   }
12 }

```

Q: 请完成 `parseQuery` 方法，要求必须使用正则

```

1  var parseQuery = function (query){
2
3  }
4
5  var obj = parseQuery("name=1&age=2");
6  console.log(obj); // { name: '1', age: '2' }

```

Q: 请完成 `concatWithSort` 方法:

```

1  var arr1 = ["A1", "A2", "B1", "B2", "C1", "C2", "D1", "D2"];
2  var arr2 = ["A", "B", "C", "D"];
3
4  function concatWithSort(arr1, arr2) {
5
6  }
7
8  var arr = concatWithSort(arr1, arr2);
9  console.log(arr); // [ 'A1', 'A2', 'A', 'B1', 'B2', 'B', 'C1', 'C2', 'C', 'D1', 'D2', 'D' ]

```

Q: 以下代码，请使用至少两种不同方法实现a与b的值互换，要求不使用临时变量，并给出解题思路

```

1  var a = 1;
2  var b = 2;

```

面试部分

Q: 请简述前后端分离架构的优缺点

Q: 请简述前端都有哪些安全问题，怎么解决

Q: 浏览器缓存读取规则

Q: 简述浏览器的工作流程

Q: **cookie** 和 **token** 都存放在 **header** 中，为什么不会劫持 **token**?

Q: 介绍模块化发展历程