

下列哪个操作是 W3C 标准定义的阻止事件向父容器传递:

正确答案: C 你的答案: 空 (错误)

```
e.preventDefault()  
e.cancelBubble=true  
e.stopPropagation()  
e.stopImmediatePropagation()
```

以下关于盒子模型描述正确的是:

正确答案: A 你的答案: 空 (错误)

标准盒子模型中: 盒子的总宽度 = 左右 margin + 左右 border + 左右 padding + width

IE 盒子模型中: 盒子总宽度 = 左右 margin + 左右 border + width

标准盒子模型中: 盒子的总宽度 = 左右 margin + 左右 border + width

IE 盒子模型中: 盒子总宽度 = width

使用 CSS 的 flexbox 布局, 不能实现以下哪一个效果:

正确答案: D 你的答案: 空 (错误)

三列布局, 随容器宽度等宽弹性伸缩

多列布局, 每列的高度按内容最高的一列等高

三列布局, 左列宽度像素数确定, 中、右列随容器宽度等宽弹性伸缩

多个宽高不等的元素, 实现无缝瀑布流布局

关于 HTML 语义化, 以下哪个说法是正确的?

正确答案: D 你的答案: 空 (错误)

语义化的 HTML 有利于机器的阅读, 如 PDA 手持设备、搜索引擎爬虫; 但不利于人的阅读
Table 属于过时的标签, 遇到数据列表时, 需尽量使用 div 来模拟表格

语义化是 HTML5 带来的新概念, 此前版本的 HTML 无法做到语义化

Header、article、address 都属于语义化明确的标签

关于 HTTP 协议, 下面哪个说法是正确的?

正确答案: C 你的答案: 空 (错误)

HTTP 协议是有状态协议。

以下是一个 Http 链接的 response 的响应头: GET /xxx/xxx/js/lib/test.js HTTP/1.1 Host: 127.0.0.1 Connection: keep-alive Pragma: no-cache Cache-Control: no-cache Accept: */*

RESTful 接口中, 利用 HTTP 协议的 method 字段来描述要对资源操作的方式, 比如 GET 表示获取资源, POST 表示新增一个资源, PUT 表示更新资源, DELETE 表示删除资源等等。

一个 HTTP 请求返回的 HTTP 状态码中, 304 表示临时重定向。

使用 `for in` 循环数组中的元素会枚举原型链上的所有属性，过滤这些属性的方式是使用 **1** 函数

参考答案 `hasOwnProperty`

在空白处填入适当的代码使输出结果成立：

```
function showMoney( ) {
```

```
};
```

```
var personA = new Object;
```

```
var personB = new Object;
```

```
personA.money= "100";
```

```
personB.money= "150";
```

```
personA.showMoney= showMoney;
```

```
personB.showMoney= showMoney;
```

输出结果：

```
personA.showMoney( ); // "100"
```

```
personB.showMoney( ); // "150"
```

你的答案 (错误)

参考答案 `return this.money;`

删除给定数组中的第二项和第三项，并且在得到的新的数组中第二项后面添加一个新的值

```
var arr1 = ['a','b','c','d','e'];
```

```
var arr2 = arr1. ( , , 'newvalue')
```

你的答案 (错误)

参考答案 `splice`

写一个求和的函数 `sum`，达到下面的效果

```
1 // Should equal 15
```

```
2 sum(1, 2, 3, 4, 5);
```

```
3 // Should equal 0
```

```
4 sum(5, null, -5);
```

```
5 // Should equal 10
```

```
6 sum('1.0', false, 1, true, 1, 'A', 1, 'B', 1, 'C', 1, 'D', 1,
```

```
7 'E', 1, 'F', 1, 'G', 1);
```

```
8 // Should equal 0.3, not 0.30000000000000004
9 sum(0.1, 0.2);
```

你的答案

- 添加笔记
-
- 求解答(13)
- 收藏
- 纠错

参考答案

```
1 function sum() {
2     var nResult = 0;
3     for (var i = 0, l = arguments.length; i < l; i++) {
4         nResult += window.parseFloat(arguments[i]) || 0;
5     }
6     return nResult.toFixed(3) * 1000 / 1000;
7 }
```

请写一个表格以及对应的 **CSS**, 使表格奇数行为白色背景, 偶数行为灰色背景, 鼠标移上去时为黄色背景。

你的答案

- 添加笔记
-
- 求解答(0)
- 收藏
- 纠错

参考答案

```
1 <table class="table">
2     <tr><td>第一行</td></tr>
3     <tr><td>第二行</td></tr>
4     <tr><td>第三行</td></tr>
5     <tr><td>第四行</td></tr>
6 </table>
7
8 <style>
9     .table tr:nth-child(2n-1) {
10         background-color:white;
11     }
12     .table tr:nth-child(2n) {
13         background-color:grey;
14     }
```

```
15     .table tr:hover{
16         background-color:yellow;
17     }
18 </style>
```

写一个 **traverse** 函数，输出所有页面宽度和高度大于 50 像素的节点。

```
1  function traverse(oNode) {
2      var aResult = [];
3      oNode = oNode || document.body;
4      if (oNode.style) {
5          var nWidth = window.parseInt(oNode.style.width, 10) || 0;
6          var nHeight = window.parseInt(oNode.style.height, 10) || 0;
7          if (nWidth > 50 && nHeight > 50) {
8              aResult.push(oNode);
9          }
10     }
11     var aChildNodes = oNode.childNodes;
12     if (aChildNodes.length > 0) {
13         for (var i = 0, l = aChildNodes.length; i < l; i++) {
14             var oTmp = aChildNodes[i];
15             aResult = aResult.concat(traverse(oTmp));
16         }
17     }
18     return aResult;
19 }
```