EXERCISE

（JavaScript Native DAY 05）

版本编号 1.0

JavaScript Native DAY05

1. 编写函数:  写一个产生[x, y]的随机数函数, 并产生10个[10, 100]的随机数，在控制台显示

运行结果如图1



图- 1

* 参考答案

此案例需要用户创建html文件, 然后HTML文档中写script标签

首先，创建随机数的函数;

然后，使用循环调用随机数函数;

最后，在控制台打印结果

实现此案例需要按照如下步骤进行。

步骤一：新建HTML文件，并添加script标签

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

</script>

</body>

</html>

步骤二：创建随机数的函数

在script标签中，创建生成随机数的函数，代码如下所示：

<script type="text/javascript">

function randomNumber(x, y) {

var num = Math.floor(Math.random() \* (y - x + 1) + x);

return num;

}

</script>

步骤三：生成10个随机数

在script标签中，使用for循环，调用10次随机数函数，并且随机数的范围为[10, 100]。代码如下所示：

console.log("产生10个随机数：");

for(var i = 0; i < 10; i++) {

console.log(randomNumber(10, 100));

}

然后运行HTML文件, 就可以在控制台打印10个随机数

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

function randomNumber(x, y) {

var num = Math.floor(Math.random() \* (y - x + 1) + x);

return num;

}

console.log("产生10个随机数：");

for(var i = 0; i < 10; i++) {

console.log(randomNumber(10, 100));

}

</script>

</body>

</html>

1. 编写函数:  求三个数的中间值的函数, 并计算10，15，20的中间值，并在控制台显示

运行结果如图2

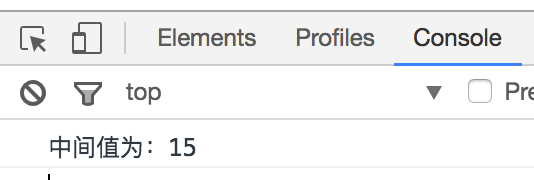


图- 2

* 参考答案

此案例需要用户创建html文件, 然后HTML文档中写script标签

首先，创建三个数中间值的函数;

然后，调用函数;

最后，在控制台打印结果

实现此案例需要按照如下步骤进行。

步骤一：新建HTML文件，并添加script标签

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

</script>

</body>

</html>

步骤二：创建三个数中间值的函数

在script标签中，创建生成三个数中间值的函数，代码如下所示：

<script type="text/javascript">

function middleValue(a, b, c) {

var max = a > b ? a : b;

max = max > c ? max : c;

var min = a < b ? a : b;

min = min < c ? min : c;

var mid = a + b + c - max - min;

return mid;

}

</script>

步骤三：调用函数

在script标签中，调用中间值函数，并打印结果。代码如下所示：

console.log("中间值为：" + middleValue(10, 15, 20));

然后运行HTML文件, 就可以在控制台打印10，15，20的中间值

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

//练习, 求三个数的中间值的函数, 并计算10，15，20的中间值

function middleValue(a, b, c) {

var max = a > b ? a : b;

max = max > c ? max : c;

var min = a < b ? a : b;

min = min < c ? min : c;

var mid = a + b + c - max - min;

return mid;

}

console.log("中间值为：" + middleValue(10, 15, 20));

</script>

</body>

</html>

1. 编写函数:  随机产生一个rgb颜色，格式为rgb(0, 255, 0)

运行结果如图3

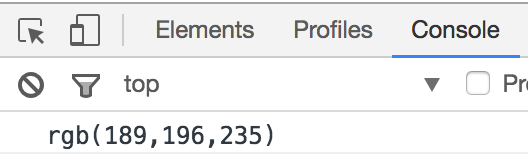


图- 3

* 参考答案

此案例需要用户创建html文件, 然后HTML文档中写script标签

首先，创建随机颜色的函数;

然后，调用随机颜色的函数;

最后，在控制台打印结果

实现此案例需要按照如下步骤进行。

步骤一：新建HTML文件，并添加script标签

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

</script>

</body>

</html>

步骤二：创建随机颜色的函数

在script标签中，创建生成随机颜色的函数，代码如下所示：

<script type="text/javascript">

function randomColor () {

var red = Math.floor(Math.random() \* 256);

var green = Math.floor(Math.random() \* 256);

var blue = Math.floor(Math.random() \* 256);

var rgbString = "rgb(" + red + "," + green + "," + blue + ")";

return rgbString;

}

</script>

步骤三：调用随机颜色函数

在script标签中，调用随机颜色函数，并在控制台打印结果。代码如下所示：

console.log(randomColor());

然后运行HTML文件, 就可以在控制台打印随机颜色

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

//练习, 随机产生一个rgb颜色，格式为rgb(0, 255, 0)

function randomColor () {

var red = Math.floor(Math.random() \* 256);

var green = Math.floor(Math.random() \* 256);

var blue = Math.floor(Math.random() \* 256);

var rgbString = "rgb(" + red + "," + green + "," + blue + ")";

return rgbString;

}

console.log(randomColor());

</script>

</body>

</html>

1. 编写函数:  实现求两个数的最大值，求三个数的最大值，求四个数的最大值

注意：该练习需要循序渐进给学员，不是一次性让学员做，可以先让学员写求两个数的最大值的函数，再写求三个数的最大值的函数，最后写求四个数的最大值的函数

* 参考答案

此案例需要用户创建html文件, 然后HTML文档中写script标签

首先，创建两个数的最大值的函数;

然后，调用函数;

再次，创建三个数的最大值的函数;

然后，调用函数;

再次，创建四个数的最大值的函数;

然后，调用函数;

最后，在控制台打印结果

实现此案例需要按照如下步骤进行。

步骤一：新建HTML文件，并添加script标签

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

</script>

</body>

</html>

步骤二：创建两个数的最大值的函数

在script标签中，创建两个数的最大值的函数，代码如下所示：

<script type="text/javascript">

function maxTwo (a, b) {

var max = a > b ? a : b;

return max;

}

</script>

步骤三：调用函数

在script标签中，调用两个数的最大值函数。代码如下所示：

console.log(maxTwo(1, 2));

然后运行HTML文件, 就可以在控制台打印两个数的最大值

步骤四：创建三个数的最大值的函数

在script标签中，创建三个数的最大值的函数，可以借助使用两个数的最大值，代码如下所示：

<script type="text/javascript">

function maxThree (a, b, c) {

//1

var max = a > b ? a : b;

max = max > c ? max : c;

return max;

//2

return maxTwo(maxTwo(a, b), c);

}

</script>

步骤五：调用函数

在script标签中，调用三个数的最大值函数。代码如下所示：

console.log(maxTwo(1, 2));

然后运行HTML文件, 就可以在控制台打印三个数的最大值

步骤六：创建四个数的最大值的函数

在script标签中，创建四个数的最大值的函数，可以借助使用两个数的最大值函数和三个数的最大值函数，代码如下所示：

<script type="text/javascript">

function maxFour (a, b, c, d) {

//1

return maxTwo(maxTwo(a, b), maxTwo(c, d));

//2

return maxTwo(maxThree(a, b, c), d);

//3

return maxTwo(maxTwo(maxTwo(a, b), c), d);

//4

var arr = [a, b, c, d];

var max = a;

for (var i = 1; i < 4; i++) {

max = max > arr[i] ? max : arr[i];

}

return max;

}

</script>

步骤七：调用函数

在script标签中，调用四个数的最大值函数。代码如下所示：

console.log(maxFour(1, 2, 3, 4));

然后运行HTML文件, 就可以在控制台打印四个数的最大值

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

//练习: 获取两个数的最大值

function maxTwo (a, b) {

var max = a > b ? a : b;

return max;

}

console.log(maxTwo(1, 2));

//练习: 求三个数的最大值

function maxThree (a, b, c) {

//1

var max = a > b ? a : b;

max = max > c ? max : c;

return max;

//2

return maxTwo(maxTwo(a, b), c);

}

console.log(maxThree(12, 21, 15));

//练习: 求四个数的最大值

function maxFour (a, b, c, d) {

//1

return maxTwo(maxTwo(a, b), maxTwo(c, d));

//2

return maxTwo(maxThree(a, b, c), d);

//3

return maxTwo(maxTwo(maxTwo(a, b), c), d);

//4

var arr = [a, b, c, d];

var max = a;

for (var i = 1; i < 4; i++) {

max = max > arr[i] ? max : arr[i];

}

return max;

}

console.log(maxFour(1, 2, 3, 4));

</script>

</body>

</html>