

Day04_函数

讲师: 聂剑峰
联系方式: 610567895

目 录

- 函数的概念及作用
- 函数的分类
- 函数的定义和调用
- 形参和实参
- 函数的作用域
- DOM的简单操作
- 函数的递归调用
- 练习和作业

➤ 函数的概念和作用

1. JS函数的概念

函数就是把特定功能的代码抽取出来，使之成为程序中的一个独立实体，起个名字（函数名）。

2. 函数的作用

正如函数的概念, 我们可以根据代码需要, 将特定的功能用函数来包裹

3. 使用函数的好处

- 1, 函数可以在同一个程序或其他程序中多次重复使用（通过函数名调用）
- 2, 使程序变得更简短而清晰
- 3, 有利于程序维护

➤ 函数的分类

函数可以分为: 内置函数 和 自定义函数

内置函数:

是官方提供好的函数, 可以直接使用
如: `alert()`, `isNaN()`, `console.log()` 等

自定义函数:

是用户自己定义的函数, 用户可以根据实际需求, 对特定的功能使用函数来封装

➤ 函数的定义和调用

1. 函数的简单定义

定义函数的语法格式:

```
function 函数名(){  
    代码块;  
}
```

注意: 1, 必须使用**function**关键字, 且为小写, 函数名可以自己给定

2, 函数名的命名规则和变量名一致

3, 函数名后必须写圆括号()

示例: 定义一个函数printOut

```
function printOut(){  
    document.write("Hello World!");  
}
```

➤ 函数的定义和调用

2, 函数的调用

函数的调用方式: 函数名()

如:

调用下面的函数: `printOut()`

```
function printOut(){  
    document.write("Hello World!");  
}
```

注意: 1, 调用函数后会执行函数内部的代码块;

2, 函数在不调用的情况下是不会执行的, 只有调用后函数中的代码才会执行

➤ 函数的定义和调用

3. 函数的标准定义

定义函数的语法格式:

```
function 函数名(参数1, 参数2, .....)  
{  
    执行语句;  
    return 返回值;  
}
```

注意: 1, 函数名后圆括号()中的参数数量不定, 也可以没有;(根据功能需要)

2, return关键字的作用是将某个值返回, 如果没有返回值则默认返回undefined;

➤ 函数的定义和调用

示例： 定义一个函数sum

```
function sum(one, two){  
    var s = one + two;  
    return s;  
}
```

函数调用:

```
var result = sum(2, 8);  
console.log(result);
```

上面代码的执行过程:

- 1, 先执行sum(2,8), 将2传给变量one, 将8传给变量two;
- 2, 函数sum内部执行one和two的相加, 然后将和返回, 返回值会赋值给result, 所以result为2和8相加后的和: 10;
- 3, 最后会打印出10

➤ 函数的定义和调用

示例练习:

- 1, 写一个函数计算1到n的和, 并返回结果打印出来; (n为函数参数)
- 2, 写一个函数计算n的阶乘, 并返回结果打印出来
- 3, 写一个函数计算两个数的最小公倍数; 并返回结果打印出来

➤ 形参和实参

形参:

形参就是在函数定义时，函数名后面的参数；

函数的形参跟变量是一样使用，且不能用var修饰

实参:

实参就是调用时，函数名后面的参数

如: 之前定义的函数sum中: one和two是形参, 2和8是实参

```
function sum(one, two){  
    var s = one + two;  
    return s;  
}  
sum(2, 8);
```

➤ 形参和实参

函数中的arguments数组:

JS中函数不介意传递进来多少个参数，也不在乎传进来参数是什么数据类型，在调用函数时也未必一定要传递指定数量的参数，原因是ECMAScript中的参数在内部是用一个数组(arguments)来表示的。函数接收到的始终都是这个数组，而不关心数组中包含哪些参数（如果有参数的话）。

arguments可以判断参数的个数，arguments是个数组(后面会详细讲解数组)。

我们可以使用arguments.length来获取参数的个数

可以使用arguments[i]的方式来访问数组中的第i个参数(i为自己给定的整数下标)

练习：向一个函数传入不定数量的数值求和。

➤ 形参和实参

注意：

在传值调用的机制中只能把实参传送给形参，而不能把形参的值反向地传送给实参。因此在函数调用过程中，当形参值发生改变，而实参中的值不会变化。

```
function addNum(n){  
    n += 5; //将形参+5, 不会改变实参  
}  
var a=10;  
addNum(a);  
console.log(a); //10
```

➤ 函数的作用域

作用域:

就是起作用的范围。或者说有效范围; 这里涉及到另外两个概念

局部变量: 定义在函数内部的变量, 这个变量只能在函数内部使用, 即作用域范围只是函数内部, 另外, 形参也是局部变量.

全局变量: 全局变量就是定义在函数外部的变量, 这个变量在任何函数中都有效, 即作用域范围是当前文件的任何地方.

注意: 在定义变量时, 如果不写关键字var也是合法的, 且是全局变量, 但是这样写不安全, 容易在其他地方被更改, 所以我们在函数中写变量要加上var

➤ 函数的作用域

示例:

```
var a = 10;  
function m1(){  
    var a = 5;  
    console.log(a); //5  
}  
m1();  
console.log(a); //10
```

- 1, 在上面函数中第一个打印为函数m1内部定义的a的值, 变量会从下往上寻找最近的变量a;
- 2, 第二个打印为最上面定义的a的值;

➤ 函数的作用域

函数的嵌套:

函数的嵌套: 函数内部可以再包含其他函数

函数之间允许相互调用, 也允许向外调用, 但是不可以调用同级函数的嵌套函数;

➤ 函数的作用域

示例练习:

- 1, 写一个函数判断一个年份是不是闰年
- 2, 写一个函数判断一个数是不是素数(又称质数,除了1和本身以外不再有其他数整除)
- 3, 年月日分别为自定义函数的参数, 判断某一个日期是否为合法的日期;
如: 2016年12月33日不是合法的日期
2016年11月13日是合法的日期

➤ DOM的简单操作

DOM的简单操作:

- 1, 获取元素节点对象: `document.getElementById('id');`
- 2, 获取输入框的内容: `value` 属性
- 3, 点击事件: `onclick`

示例:

在输入框中输入数字, 点击按钮调用函数来判断奇偶性?

注意: `document.write()` 在文档页面加载完后使用会覆盖页面内容, 尽量少用

➤ DOM的简单操作

事件驱动:

因为函数不会主动执行,只有当调用的时候才会执行函数中的代码,在前面的dom操作示例中,我们点击了按钮才触发函数调用

所有的函数,没有调用不会执行,那么在我们浏览器中函数调用的源头在何处,就是事件,只有使用事件去驱动,函数才被调用;如: **onclick: 点击事件**

示例: 点击搜索按钮调用函数打印出文本框内容,再清空文本框的内容

➤ 函数的递归调用

递归调用:

函数可以自己调用自己,成为函数的递归调用;

重要性:

递归的分量,递归属于函数中比较难理解的知识,在应用开发中,虽然使用不是很频繁,但是很体现你的功底,而且,从事IT行业开发,最好要会递归,如果说现在可以不要求灵活运用的话,以后到公司中一定要会,如果面试中有人问你递归,说明,他对你要求挺高.

递归调用的方式:

1. 首先去找临界值,即无需计算,获得的值(一般是返回该值)。
2. 找这一次和上一次的关系(一般从后往前找)
3. 假设当前函数已经可以使用,调用自身计算上一次的运行结果,再写出这次的运行结果。

看上去还是很迷茫,我们来举个例子:

➤ 函数的递归调用

示例:

5的阶乘是多少?

5的阶乘=5 * 4 * 3 * 2 * 1

4 的阶乘= 4 * 3 * 2 * 1

3 的阶乘= 3 * 2 * 1

2 的阶乘= 2 * 1

1的阶乘= 1

0的阶乘= 1

1, 临界条件: $1! = 1$
2, n 的阶乘= $n * (n-1)$ 的阶乘;

➤ 函数的递归调用

练习:

用递归实现:

1, 计算 $1+2+3+\dots+100$ 的和

2, 输入一个 n , 打印 n 个hello world

➤ 作业

- 1, 课堂代码写一遍, 掌握函数的定义, 调用, 能够灵活使用函数
- 2, 写个函数计算所有传入参数相乘的结果(参数数量不定)
- 3, 写个函数实现加减乘除运算(两个操作数, 一个操作符)
- 4, 写个函数实现n的阶乘
- 5, 找出0-100之间7的倍数, 和包含7的数字
- 6, 不用循环, 计算0-100以内的数字和
- 7, 制作一个简易计算器(计算两个数的加减乘除, 带可手动输入数字和运算符)
- 8, 兔子繁殖问题, 设有一对新生兔子, 从第四个月开始他们每个月, 月初都生一对兔子, 新生的兔子从第四个月月初开始又每个月生一对兔子按此规律, 并假定兔子没有死亡, $n (n \leq 20)$ 个月月末共有多少对兔子?

THANKS