logo.png

javascript基础四

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# 一、函数

## 1.1 什么是函数

函数就是可以重复执行的代码块。

为了完成特定功能。

## 1.2为什么要用函数

因为一部分代码使用次数可能会很多，所以封装起来，需要的时候调用就可以了。

以后任何时候写代码，首先考虑，有没有什么功能是需要封装的，再一个有没有什么功能是以前封装好的。

面向对象编程是一种编程思想。特点：封装、继承、多态。

# 二、参数

参与运算的变量。

为了增强函数的功能性，和程序员的交互性，和函数的可拓展行。

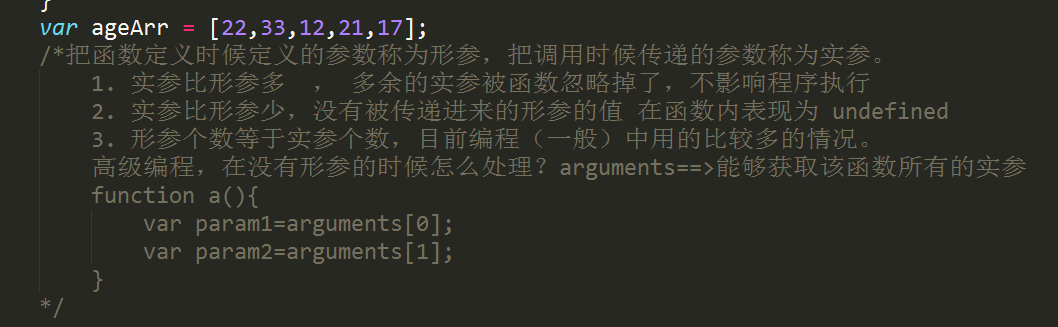
## 2.1 形参

形式上参与运算的变量，无实际值，为实参占位置。（可以理解为函数的内部变量外部无法访问）

## 2.2 实参

实际参与运算的变量。形参为他占位置，真实参与运算的变量。

注意；js没有方法重载 如果存在参数不匹配



# 三、返回值

函数程序运行后的结果外部需要使用的时候，我们不能拿到，需要通过return返回。

函数内部，return后面的值就是返回值；

### 注意

1. 函数使用return语句后，这个函数会在执行完 return 语句之后停止并立即退出，也就是说return后面的所有其他代码都不会再执行。

## 案例

1、求圆的周长和面积 Math.PI = 3.1415…

2、求2个数中的最大值，求3个数中的最大值

调用已经封装好的方法 来处理新的需求

3、求一组数中的最大值和最小值

~~查找数组中是否存在 二分法~~

4、翻转数组，返回一个新数组

根据下标调整位置

5、对数组排序，从小到大

6、求阶乘

递归思想 （遍历文件夹、tree结构）

7、求1!+2!+3!+....+n!

8、判断一个数是否是素数

## 函数名、函数体和函数加载问题（重点记忆）

1、函数名是什么

函数名 == 整个函数。

function fn(){alert(1)};

console.log(fn) == console.log(function fn(){alert(1)});

2、三种定义函数的定义

2.1 声明 function a() {}

2.2 匿名函数 var a = function(){…}

~~2.3 var a = new Function(‘var a = “name”,console.log(a)’) ; a();~~

## 作业：

1. 求斐波那契(宇宙的真理！)数列Fibonacci中的第n个数是多少？

有一对兔子，从出生起后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子， 假如兔子都不死，问第二十个月的兔子对数为多少？ 不死神兔

1 1 2 3 5 8 13 21...

记住怎么实现，可能会是面试题

递归的思想 简化代码 和工作量

注意：递归一定要有跳出条件

1. 输入某年某月某日，判断这一天是这一年的第几天？（闰年）

（四年一闰，百年不闰，四百年在闰）

只有在闰年的时候366天 在2月29天多一天

其余的时间2月都是28

1. 是否是闰年
2. 每个月多少天

# 四、变量和作用域（隐式全局变量和变量声明提升）

## 4.1变量作用域

1. 全局变量（成员变量）

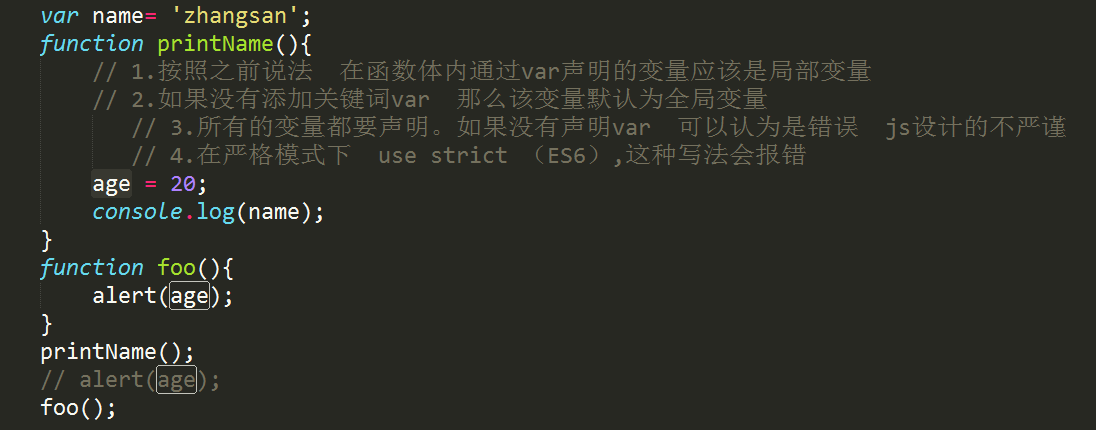
哪里都可以访问到的变量。

1. 局部变量

函数内部的变量，只有函数内部可以访问到。

（函数内部用var定义的变量和形参）

注意： 没有var声明的变量



## 4.2 变量提升

JavaScript的函数作用是指在函数内声明的所有变量在函数体内始终是有定义的，也就是说变量在声明之前已经可用，所有这特性称为声明提前（hoisting）

|  |
| --- |
| var scope = 'global';  function f(){  console.log(scope);  var scope = 'local';  console.log(scope); } // 以上代码在js进行解析的时候会变成以下代码  function f(){  var scope;  console.log(scope);  scope = 'local';  console.log(scope);  }  f(); |

## 4.3 函数声明提升

|  |
| --- |
| var getName = function(){  console.log(2);  }  function getName (){  console.log(1);  }  getName(); |

## 4.4 ~~隐式全局变量(不提倡这么写，在严格模式下是报错的)~~

隐式全局变量就是隐藏的全局变量不好被发现。

function fn（）{

var a = b = c = 1; // b和c就是隐式全局变量

}

注意:

function fn（）{

var a = b = c = 1; // b和c就是隐式全局变量（等号）

var a = 1; b = 2; c = 3; // b和c就是隐式全局变量（分号）

var a = 1 , b = 2 , c = 3; // b和c就不是隐式全局变量（逗号）

}

# 五、总结：

* 函数不调用不执行
* 函数名就等于（整个函数）
* 参数相当于局部变量
* 就近原则使用变量
* 两个平级的函数中的变量不会相互影响（可以使用同样的形参名）

# 六、函数高级

## 匿名函数（了解）

定义：匿名函数就是没有名字的函数。

作用：

1. 不需要定义函数名的时候。
2. 书写起来更简便。

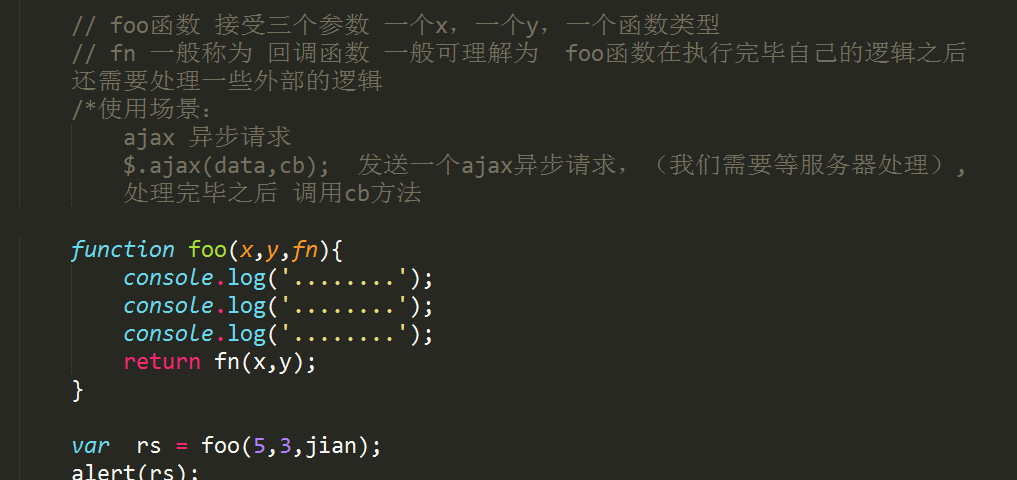
匿名函数的执行方式：()()

什么时候使用：假如一个函数不会被其他地方调用，同时又想让其执行，就使用()()使函数立即执行。

## 函数是一种类型（了解）

function 类型 函数实际上是Function(对象)的一个实例

## 函数作为参数（了解）ajax



## 递归（理解）

必须有跳出条件。

# 七、JS中的对象（Object）

Java/c js 会有一个Object对象 ，是一个基类。

所有的对象都是继承自Object对象的

## 7.1 创建空白对象

var obj = new Object();

## 7.2 构造函数（就是为了创建对象实例）

可以创建对象实例的函数。

区别与普通函数，首字母大写。

构造函数实际上是做了一些操作

## 7.3 创建自定义函数

## 7.3.1 This

this只出现在函数中。 （如果学过java，跟java要区分开）

面试题：js中 this指的是谁？

记住：this指向调用者。

new People(); People中的this代指被创建的对象实例。

## 7.3.2 new

1.开辟内存空间，存储新创建的对象（ new Object() ）

2.把this设置为当前对象

3.执行内部代码，设置对象属性和方法

4.返回新创建的对象