昨日回顾

流程控制：

分结构：

if

switch

循环结构：

for(初值;终值;步长){

循环体

}

while(表达式){

}

for 用于判断循环次数已知

while 用于循环次数未知

do…while() 行执行循环体再判断表达

循环结束与退出

continue【n】

结束当前循环结构的本轮循环，继续上n层循环结构的下一次循环。

break【n】

直接结束上n层循环结构。

文件的包含：

一个php文件会引入其他文件：

php会引入html：

将当前Php中的数据，显示在一个html文档，最终将动态的数据与静态的html文档混合一起，返回给浏览器。html文件模板文件。

被引入的html文档中也会引入其他的静态资源文件(css、image、js)，此时静态资源文件的路径需要与php当前目录为相对路径。

php会引入php文件：

在实际开发过程中，功能性的代码会单独存放在一个php文件，如果其他文件想使用这个功能，则需要引入。

require()

require\_once()

当被引入的文件不存时，会中断程序的执行。

require\_once() 引入文件时会检查被引入的文件是否引入过。

include()

include\_once()

当被引入的文件不存时，不会中断程序的执行。

include\_once() 引入文件时会检查被引入的文件是否引入过。

PHP错误机制

错误：分两大类

语言的错误： 违反语言语法规则。例如：关键字写错误，没有分号，系统函数所必须的参数没有传递。

编译错误

E\_PARSE

执行错误

E\_NOTICE

E\_WARNING

E\_ERROR

程序员认为的错误、程序员逻辑错误：

function showInfo($arr){

trigger\_error(‘错误信息’,); 本质就是将错误信息写到一个文本文件中。

}

E\_USER\_ERROR

E\_USER\_NOTICE

E\_USER\_WARNING

一、函数初步

1、基本概念

函数是模块化程序的产物。在实际开发过程中，我们会将一个项目划分为各个大的功能模块。同将大的功能模块发为小的功能。再使用函数将这个最小的功能(一个动作)实现。

从代码上看，函数就是使用{}，将完成某一功能的代码括起来(封装)，再使用function 关键字，为这段代码设置一个名子，这个名子就是函数名，函数名后必须有()，这个()表示函数只能被调用。

2、声明语法

语法：

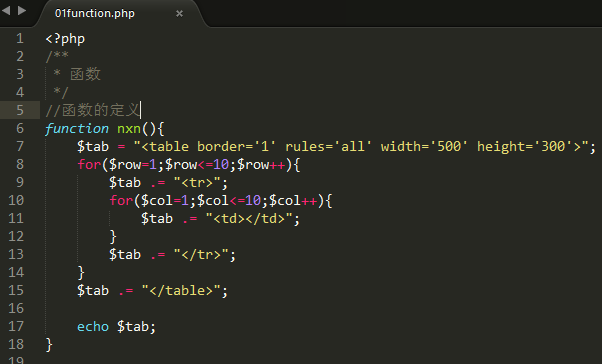
function funName(形参列表){

函数体

【return 值】

}

示例：

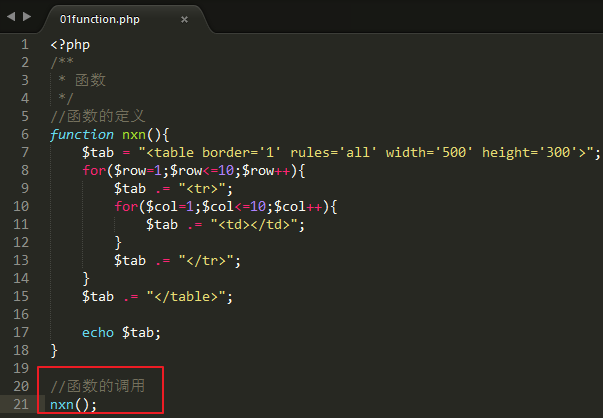


3、调用语法

语法：

funName(实参列表);

示例：



4、函数名

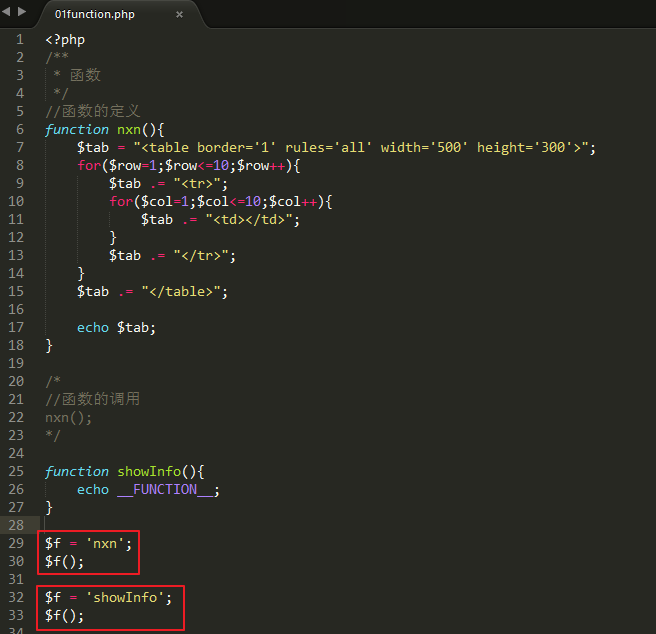
php中函数名是不区分大小写。函数名的命名规则与变量的命名规则相同。只能包含字母、数字、下划线。而且以字母下划线开头。

5、可变函数

如果$a的值是另一个变量的名，那么可以使用$$a来访问另一个变量的值。

如果$f变量的值是另一个函数名，那么可以使用这个变量来访问函数

示例：



示例：



6、函数体

完成某一功能代码段

7、return语句

在我们使用函数完成一些功能时，有时会将函数计算的数据，传递出来。此时可以使用return。

提示：

程序是用于处理数据，函数所处理的数据有两个方向：

其一，从外到内，使用实参与形参。

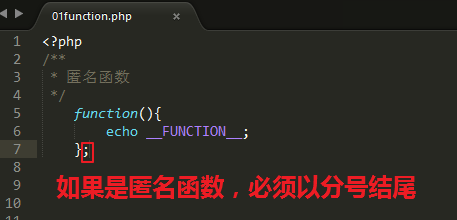
其二，从内到外，使用return

函数内部可以有多个return，只要遇到一个return就会中断函数的执行。

函数内也可省略return语句。

8、匿名函数

没有名子的函数就是匿名函数(anonymous function)，也称之为闭包函数(closure)。

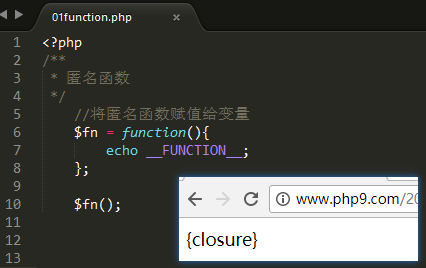


提示：

JS中的函数可以自调用，但是PHP中的函数没有办法自调用。

php中的匿名函数，可以赋值给一个变量，还可以用于某个函数的参数。

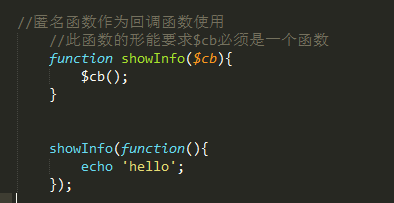
示例：将匿名函数赋值给变量，通过这个变量可以调用匿名函数。



回调函数callback：

我们定义一个函数很多的时候是为了自己调用。但在开发过过程中，我们在某别人的函数，或系统函数时。这个些函数需要参数，而有些参数要求你传递的一个函数。 。

示例：匿名函数作为回调函数(参数)传递



二、函数的参数

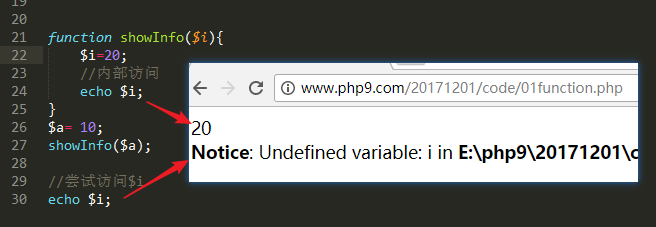
1、形参

函数定义时的参数就是形参

形参是被动的接收数据。

形参只对内存有效，对于外部是不可见。

示例：



2、实参

函数调用时的参数就是实参

主动向函数内部传递实际的数据。

提示：

形参与实际让函数更具有通用性。

3、形参的默认值

php中在定义函数时可以为形参赋值一个值，这个值就是形参的默认值。

在调用函数时，如没有为具有默认值的形参传递数据，那么形参使用的是默认值，

如果有为具有默认值的形参传递数据，那么这个形参使用实际传递过来的数据。

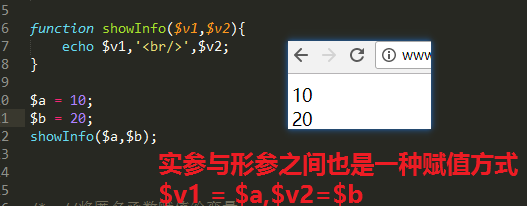
一般具有默认值的形参位于，形参列表的最后。

示例：



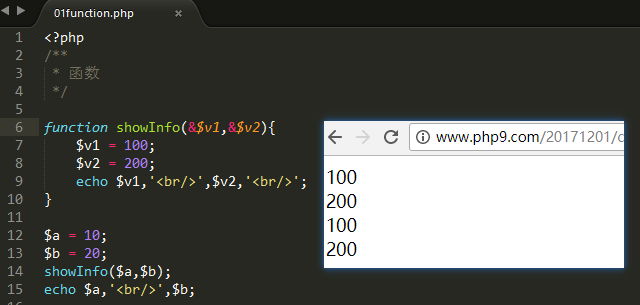
4、形参的引用传值

php中函数的实参与形参之间也是一种赋值方法，默认是





PHP中可以在形参前加&符号，将实参与形参之间默认的赋值传值，更改为引用传值。



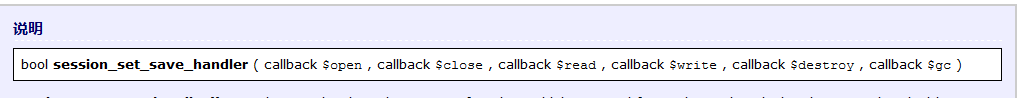
5、伪类型(函数)

PHP语言本身提供了8种数据类型。但是在使用手册中我们会遇到另外几中：

mixed 表示类型不确定

callback 表示函数

scalar 如果是int、float、string、bool



6、相关函数(系统函数)

func\_get\_args();

用于获取实参，并以数组的形式返回。

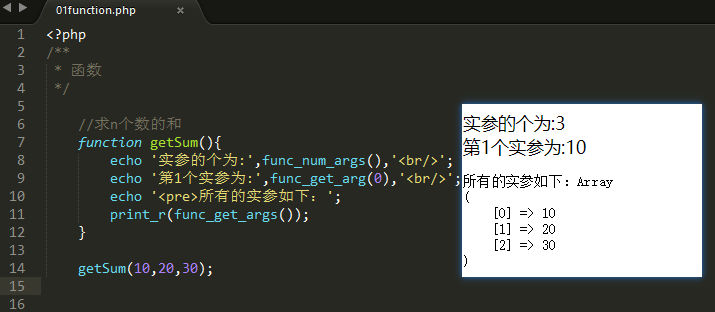
func\_get\_arg(ind);

用于获取ind下标指定的实参

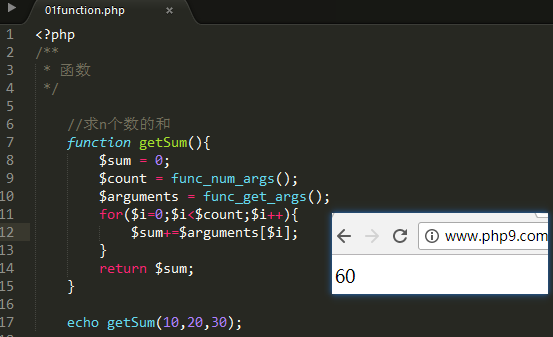
func\_num\_args();

用于获取实参的个数。

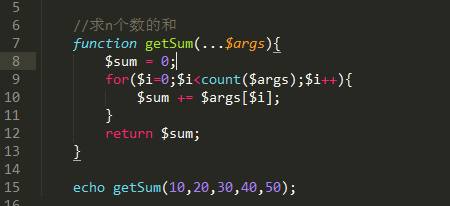
示例：



示例：应用



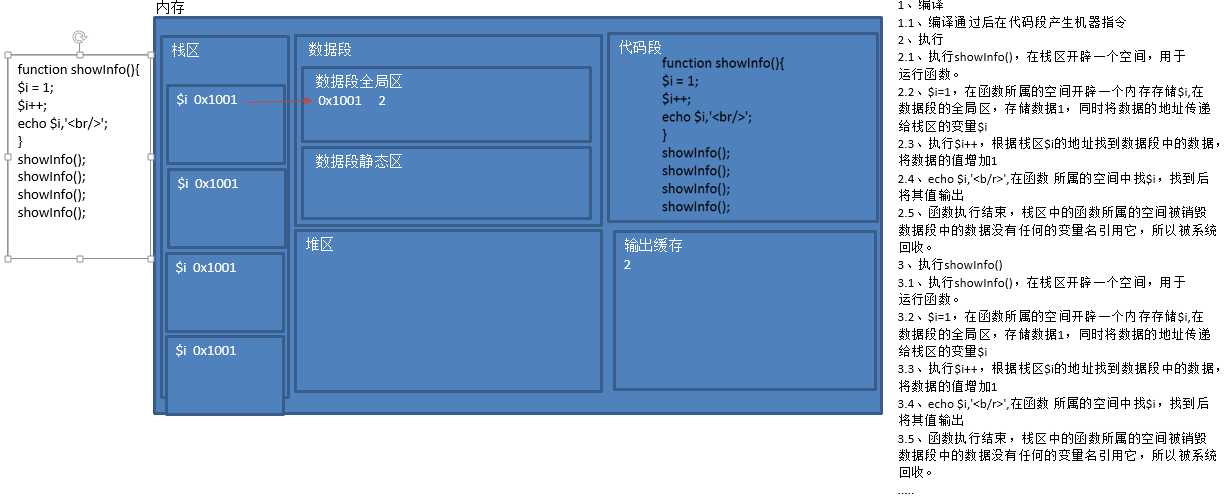
示例：



说明：

…$变量名 用于将实参以数组元素的形式保存在这个变量中。

7、函数运行的内存原理



三、作用域

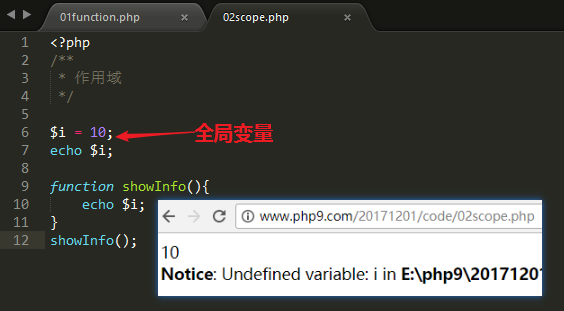
在PHP中函数是用于封装某一功能的代码段，函数在对代码进行封装会有一些影响。

PHP中分两种作用域：全局作用域，局部作用域。

1、全局作用域与全局变量

在函数外部的定义的变量，其作用域 就是全局作用域，变量就是全局变量。

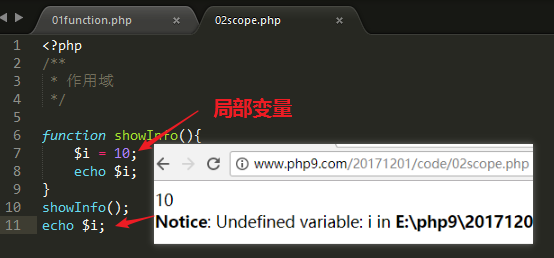
示例：



2、局部作用域与局部变量

在函数内部定义的变量，其作用域就是局部作用域，这个变量就是局部变量。

示例：



提示：

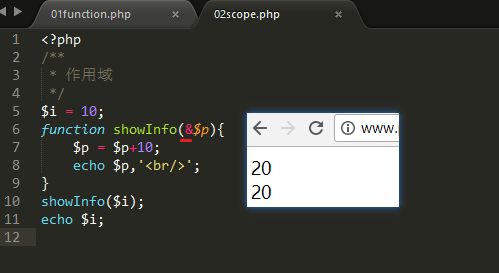
默认php中有很”清晰”的作用域，外部只能访问外部的，内部只能访问内部。

3、global关键字

虽然php默认作用很清晰，但是有时确定会外部访问内部或内部访问外部。如何解决？

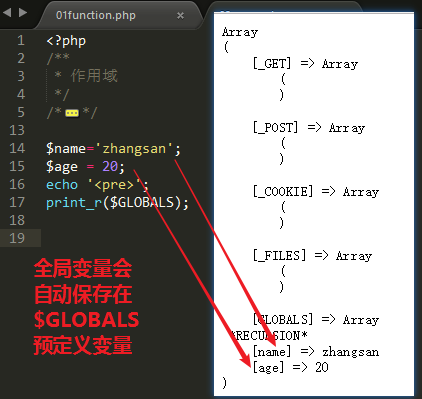
①、通过参数的传址方式

在形参前使用&，可以达到内部访问外部

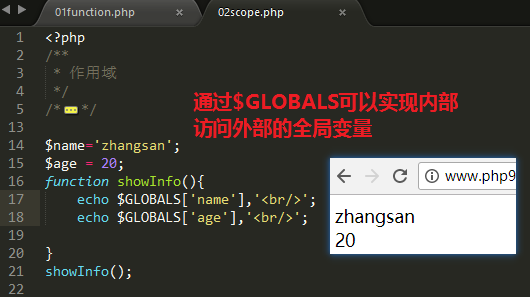


②、$GLOBALS

$\_GET\$\_POST….$GLBOALS，超全局变量



示例：



提示：  
 以上两种变通的方式都只是实现了内部访问外部，如果我即想内部访问外部，外部也同时可以访问内部。

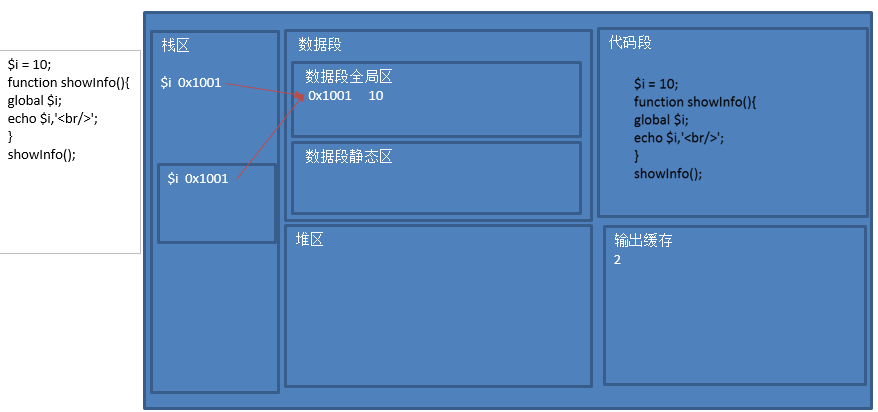
③、global关键字

语法:

global 变量名;

说明：

在函数内部建立一个与函数外部同名的变量的引用。如果外部有这样的同名变量，会在外部创建一个同名的变量。



4、常量的作用域

常量没有作用域限制，一个脚本中的常量，在任何的位置都可以访问。

5、预定义变量(超全局变量)

超全局变量也不受作用域的限制。

四、变量的生命周期

1、作用域与生命周期

作用域 表示的是一个变量的作用空间范围。

生命周期 表示的是一个变量的作用时间范围。

2、全局变量的生命周期

全局变量 生命周期是从脚本执行开始到脚本执行结束。

3、局部变量的生命周期

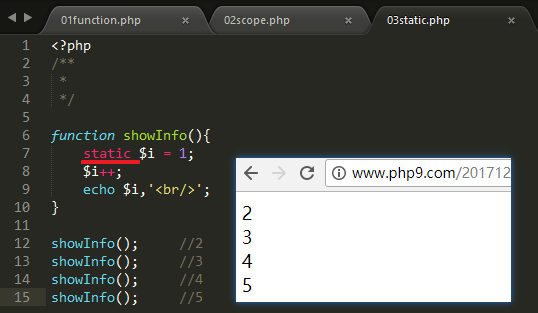
局部变量 生命周期是从函数执行开始，到函数执行结束。

五、 静态变量

1、定义

在函数内部使用static声明的变量，就是静态变量。

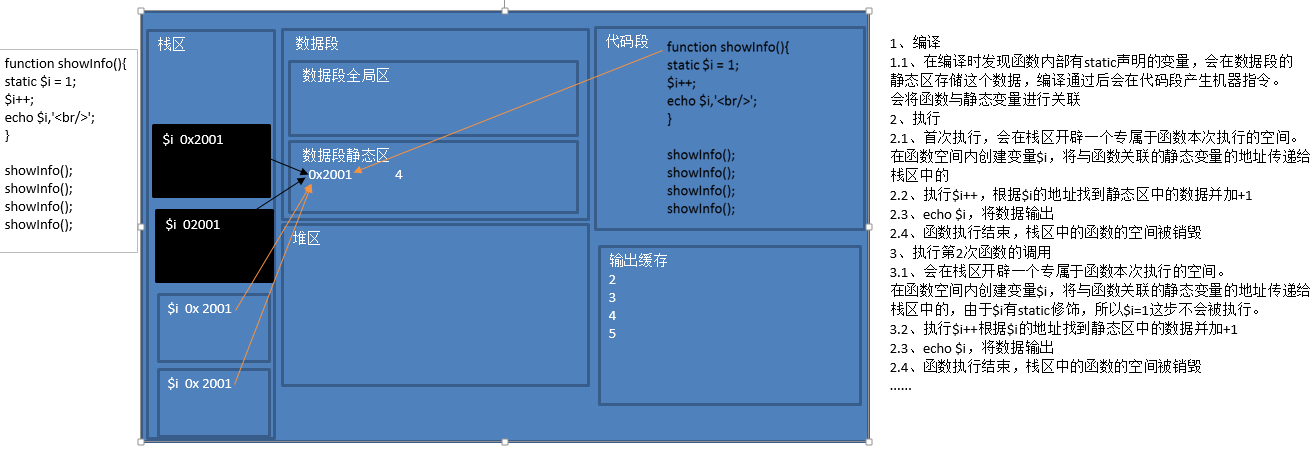
示例：



2、原理

静态变量在函数多次被调用时，只会被初始化一次，并且静态变量的值并不会随着函数执行后空间的销毁而被销毁。

在函数下一次调用时，仍然可以访问其值。



提示：

静态变量的使用场景，如果想在同一个函数多次调用时，共享一份数据。那么就使用静态变量。

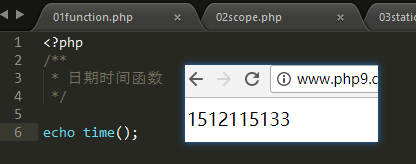
六、系统函数

1、日期时间函数

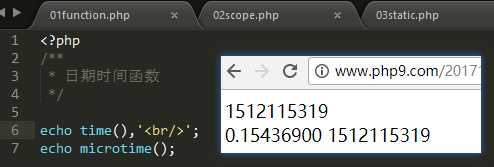
①、time()

说明：  
 用于获取当前时间的时间戳。单位是秒。时间戳就是从时间原点至现在的一个秒数。

时间原点，计算机中规定从1970年1月1日0时0分0秒那一时刻为时间原点。



②、microtime();



③、date()

语法：

date(format【,time】);

说明：

用于格式化时间信息。

time表示所要格式化的时间戳，如果省略表示对当前时间进行格式化。

format用于格式化时间的字符串

Y 表示4位的年份

m 月份，有前导0

d 表示，日期

H 表示24小时机制的小时数

i 表示分钟数

s 表示秒数

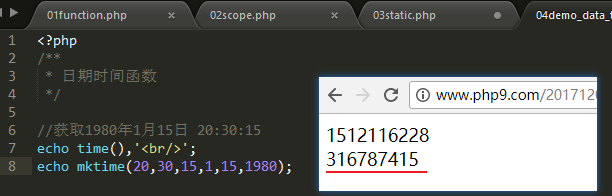
④、mktime()

语法：

mktime(时,分,秒,月,日,年);

说明：

用于获取指定时间的一个时间戳。



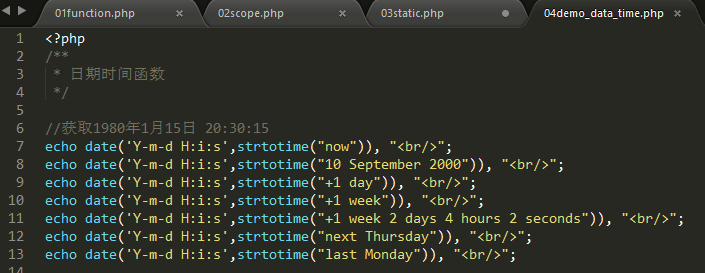
⑤、strtotime()

语法：

strtotime(string)

说明：

将一个以字符的形式描述的时间信息转换为时间戳。



七、递归(了解)

函数自己调用自己就是递归。

递归解决问题的特点：完成一个需求有多个步骤，但完成每一步所实现的代码是十分相似的，只有函数所传递的参数不同而已。

示例：



fibonacci数列