常用词汇记录：

关于错误调试：

Error\_reporting() 括号中写0表示屏蔽掉所有的错误而不显示

\_\_FILE\_\_ ：表示的是当前文件的绝对路径

关于&（引用变量的使用）

引用变量并不是复制一个变量给另一个变量，而是将变量指向同一个内容，可以理解为将同一个变量的地址传递给另一个变量。引用后，两个变量完全相同。当对其中的任意一个变量的内容进行更改的时候，另一个变量的内容也会随之而发生更改

引用和复制的区别在于：

复制是将原变量内容复制下来，开启一个新的内存空间来保存，而引用则是给变量的内容在命名一个名字

、

变量的作用域：

变量按照作用域分为全局变量，局部变量，静态变量，

全局变量全局通用，使用关键字global进行声明，或者通过全局数组4global进行访问

局部变量： 在函数内部定义的变量，只能在函数内部使用，在函数外部不能够被使用

静态变量： 能够在函数调用结束之后仍保留变量值，当再次回到作用域的时候，又可以继续使用原来的值，一般变量函数调用结束之后，存储的值就会被清空，所占的内存空间也会被释放掉。使用静态变量的时候，首先是用static来声明变量，需要把关键字static放在要定义的变量的前面、

声明格式如下：

Global $str;

Static $str;

示例如下：

<?php

function zdy(){

static $message = 0; //初始化静态变量

$message += 1; //静态变量+1;

echo $message; //输出静态变量

}

function zdy1(){

$message = 0; //声明函数内部变量（局部变量）

$message += 1; //局部变量+1

echo $message.""; //输出局部变量

}

for($i=0;$i<10;$i++){ //输入1-10

zdy();

}

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

zdy1(); //输出10个1

}

/\*

如上所示：

我们知道了静态变量在函数调用结束之后依然会保留值，普通的变量就不会保留

\*/

?>

值得注意的是：

默认情况下全局变量和局部变的作用域是不相交的，所以，在函数内部可以定义与全局变量同名的变量。全局变量可以在程序中的任何地方访问，但是在用户自定义函数内部是不可用的，想在用户自定义函数内部使用全局变量，要是用关键字global进行声明

关于$ （可变变量）

可变变量是一种独特的变量，变量的名称不是预先定义好的，而是动态的设置和使用，可变变量一般指使用一个变量的值作为另一个变量的名称，所以可以变量又称为变量的变量，可变变量通过在一个变量名称前使用两个“$” 符号实现

示例如下：

<?php

$str = "bbb";

$bbb = "abcd";

echo $$str;

?>

引用文件的方法：

Require() 🡪 该语句相当于将另一个源文件的内容完全复制到文本中，所以一般将源文件放置到开位置，用于引用需要的公共函数文件和公共类文件等

Include once() 他会在引用以来去检测该文件是否被引用过，如果有，就不会再次被引用了，如果重复导入这个文件，第二次导入就会发生错误，因为php中不允许相同名称的函数被重复引用

Require once() 他和上着的区别就是上面的一个重复就会报错，而现在的这个第二次引用不会报错，。但是第二次的引用不会被执行

上面两个语句的重用是：确保被包含的文件只能被包含一次，使用这种语句可以防止意外的多次包含相同的函数库，从而减少函数重复定义并产生错误。

Include\_\_once 语句在脚本执行期间调用外部文件发生错误的时候，产生一个警告，而require\_\_once语句则导致一个致命的错误

Include 和 require 的区别：

Require语句调用文件，如果调用的文件没有找到，他就会报错，并且立即终止脚本的处理，

只要程序一执行，就会立刻调用外部文件，

Include语句调用文件，没有找到文件就会立即输出警告，但是不会终成宿的执行

通过include调用外部文件的时候，只有程序执行到该语句时，才会调用外部文件

截取字符串： 第一个表示需要截取的字符串，第二个表示从什么位置开始截取，第三个表示截取字符串的长度

substr($str,1,1)、

随机数 ：第一个为最小值，第二个参数为最大值

Rand（0,1）

Foreach遍历：第一个参数为需要遍历的数组，第二个将遍历数值的值指向他

Foreach（$value as $value2）{

}

示例：

<?php

$str = array(

array(1,2,3,4),

array(1,2,3,4),

array(1,2,3,4),

array(1,2,3,4)

);

foreach($str as $aa){

foreach($aa as $bb){

echo $bb;

}

}

?>

结果为：

