# 基本语法

## PHP标记

起始标记：<?php

结束标记：?>

这样写可以将php代码嵌入到任何的文档中

PHP也有短标记<? ?>,不鼓励；只有在php.ini中打开short\_open\_tag或者在编译是使用配置选项



如果文件内容是纯PHP代码，最好在文件末尾删除结束标记

版本5.4以后，增加了段标记：<?=

每个语句后面要用分号结束，如果是最后一行可以省略分号。

//最后一行可以不用使用分号，但要有php的结束标记

<?php

phpinfo()

?>

//如果没有结束标记，需要分号

<?php

phpinfo();

为什么不需要文件末尾的PHP结束标记：

文件末尾的 PHP 代码段结束标记可以不要，有些情况下当使用 [include](mk:@MSITStore:F:\Html\mannual_files_all\enhancingProgram\PHP\php_manual_zh_review.chm::/res/function.include.html) 或者 [require](mk:@MSITStore:F:\Html\mannual_files_all\enhancingProgram\PHP\php_manual_zh_review.chm::/res/function.require.html) 时省略掉会更好些，这样不期望的空白符就不会出现在文件末尾，之后仍然可以输出响应标头。在使用输出缓冲时也很便利，就不会看到由包含文件生成的不期望的空白符。

单行注释：// 或者 #

多行注释:/\* \*/,不能嵌套

PHP支持8中原始数据类型：

Boolean integer float/double string

Array object

Resource Null

伪类型：Mixed number callback

PHP中变量的类型通常不是由程序员设定的，是由PHP根据该变量使用的上下文在运行时决定的。

查看类型：var\_dump 或者 gettype 或者判断类型is\_null is\_string等

<?php

$str = 'abc';

$num = 12;

$nil = null;

var\_dump($str);

var\_dump($num);

var\_dump($nil);

echo gettype($str) . "\n";

echo gettype($num) . "\n";

echo gettype($nil) . "\n";

var\_dump(is\_int($str)) . "\n";

类型强制转换：settype 和各种前缀（int bool float string array object unset-转换为null）

<?php

//$str='addaa'; #强制转换为0

$str='12';

if(settype($str,'integer')){

var\_dump($str);

}else{

echo '强制转换失败';

}

### Boolean类型：

True和false关键字不区分大小写

<?php

$bol = TRUE;

$bol = false;

$bol = truE;

var\_dump($bol);