# PHP 教程

PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言。

PHP 是免费的，并且使用非常广泛。同时，对于像微软 ASP 这样的竞争者来说，PHP 无疑是另一种高效率的选项。

# PHP 简介

PHP 是服务器端脚本语言。

## 您应当具备的基础知识

在继续学习之前，您需要对以下知识有基本的了解：

* HTML
* CSS

如果您希望首先学习这些项目，请在我们的 [首页](https://www.runoob.com/) 访问这些教程。

## PHP 是什么？

* PHP（全称：PHP：Hypertext Preprocessor，即"PHP：超文本预处理器"）是一种通用开源脚本语言。
* PHP 脚本在服务器上执行。
* PHP 可免费下载使用。

|  |  |
| --- | --- |
| **IMG_256** | PHP 对初学者而言简单易学。PHP 也为专业的程序员提供了许多先进的功能。 |

## PHP 文件是什么？

* PHP 文件可包含文本、HTML、JavaScript代码和 PHP 代码
* PHP 代码在服务器上执行，结果以纯 HTML 形式返回给浏览器
* PHP 文件的默认文件扩展名是 ".php"

## PHP 能做什么？

* PHP 可以生成动态页面内容
* PHP 可以创建、打开、读取、写入、关闭服务器上的文件
* PHP 可以收集表单数据
* PHP 可以发送和接收 cookies
* PHP 可以添加、删除、修改您的数据库中的数据
* PHP 可以限制用户访问您的网站上的一些页面
* PHP 可以加密数据

通过 PHP，您不再限于输出 HTML。您可以输出图像、PDF 文件，甚至 Flash 电影。您还可以输出任意的文本，比如 XHTML 和 XML。

## 为什么使用 PHP？

* PHP 可在不同的平台上运行（Windows、Linux、Unix、Mac OS X 等）
* PHP 与目前几乎所有的正在被使用的服务器相兼容（Apache、IIS 等）
* PHP 提供了广泛的数据库支持
* PHP 是免费的，可从官方的 PHP 资源下载它：[www.php.net](http://www.php.net/" \t "https://www.runoob.com/php/_blank)
* PHP 易于学习，并可高效地运行在服务器端

# PHP 语法

PHP 脚本在服务器上执行，然后将纯 HTML 结果发送回浏览器。

## 基本的 PHP 语法

PHP 脚本可以放在文档中的任何位置。

PHP 脚本以 **<?php** 开始，以 **?>** 结束：

<?php  
// PHP 代码  
?>

PHP 文件的默认文件扩展名是 ".php"。

PHP 文件通常包含 HTML 标签和一些 PHP 脚本代码。

PHP 中的每个代码行都必须以分号结束。分号是一种分隔符，用于把指令集区分开来。

通过 PHP，有两种在浏览器输出文本的基础指令：**echo** 和 **print**。

# PHP 变量

变量是用于存储信息的"容器"：

## 实例

<?php $x=5; $y=6; $z=$x+$y; echo $z; ?>

**[运行实例 »](https://www.runoob.com/try/showphp.php?filename=demo_var1" \t "https://www.runoob.com/php/_blank)**

## 与代数类似

x=5  
y=6  
z=x+y

在代数中，我们使用字母（如 x），并给它赋值（如 5）。

从上面的表达式 z=x+y ，我们可以计算出 z 的值为 11。

在 PHP 中，这些字母被称为**变量**。

|  |  |
| --- | --- |
| **IMG_256** | 变量是用于存储数据的容器。 |

## PHP 变量

与代数类似，可以给 PHP 变量赋予某个值（x=5）或者表达式（z=x+y）。

变量可以是很短的名称（如 x 和 y）或者更具描述性的名称（如 age、carname、totalvolume）。

PHP 变量规则：

* 变量以 $ 符号开始，后面跟着变量的名称
* 变量名必须以字母或者下划线字符开始
* 变量名只能包含字母数字字符以及下划线（A-z、0-9 和 \_ ）
* 变量名不能包含空格
* 变量名是区分大小写的（$y 和 $Y 是两个不同的变量）

|  |  |
| --- | --- |
| **IMG_257** | PHP 语句和 PHP 变量都是区分大小写的。 |

## 创建（声明）PHP 变量

PHP 没有声明变量的命令。

变量在您第一次赋值给它的时候被创建：

## 实例

<?php $txt="Hello world!"; $x=5; $y=10.5; ?>

**[运行实例 »](https://www.runoob.com/try/showphp.php?filename=demo_var2" \t "https://www.runoob.com/php/_blank)**

在上面的语句执行中，变量 **txt** 将保存值 **Hello world!**，且变量 **x** 将保存值 **5**。

**注释：**当您赋一个文本值给变量时，请在文本值两侧加上引号。

## PHP 是一门弱类型语言

在上面的实例中，我们注意到，不必向 PHP 声明该变量的数据类型。

PHP 会根据变量的值，自动把变量转换为正确的数据类型。

在强类型的编程语言中，我们必须在使用变量前先声明（定义）变量的类型和名称。

## PHP 变量作用域

变量的作用域是脚本中变量可被引用/使用的部分。

PHP 有四种不同的变量作用域：

* local
* global
* static
* parameter

## 局部和全局作用域

在所有函数外部定义的变量，拥有全局作用域。除了函数外，全局变量可以被脚本中的任何部分访问，要在一个函数中访问一个全局变量，需要使用 global 关键字。

在 PHP 函数内部声明的变量是局部变量，仅能在函数内部访问：

在以上实例中 myTest() 函数定义了 $x 和 $y 变量。 $x 变量在函数外声明，所以它是全局变量 ， $y 变量在函数内声明所以它是局部变量。

当我们调用myTest()函数并输出两个变量的值, 函数将会输出局部变量 $y 的值，但是不能输出 $x 的值，因为 $x 变量在函数外定义，无法在函数内使用，如果要在一个函数中访问一个全局变量，需要使用 global 关键字。

然后我们在myTest()函数外输出两个变量的值，函数将会输出全局变量 $x 的值，但是不能输出 $y 的值，因为 $y 变量在函数中定义，属于局部变量。

|  |  |
| --- | --- |
| **IMG_256** | 你可以在不同函数中使用相同的变量名称，因为这些函数内定义的变量名是局部变量，只作用于该函数内。 |

## PHP global 关键字

global 关键字用于函数内访问全局变量。

在函数内调用函数外定义的全局变量，我们需要在函数中的变量前加上 global 关键字：

PHP 将所有全局变量存储在一个名为 $GLOBALS[*index*] 的数组中。 index 保存变量的名称。这个数组可以在函数内部访问，也可以直接用来更新全局变量。

## Static 作用域

当一个函数完成时，它的所有变量通常都会被删除。然而，有时候您希望某个局部变量不要被删除。

要做到这一点，请在您第一次声明变量时使用 **static** 关键字：

然后，每次调用该函数时，该变量将会保留着函数前一次被调用时的值。

**注释：**该变量仍然是函数的局部变量。

## 参数作用域

参数是通过调用代码将值传递给函数的局部变量。

参数是在参数列表中声明的，作为函数声明的一部分：

* 1、定义在函数外部的就是全局变量，它的作用域从定义处一直到文件结尾。
* 2、函数内定义的变量就是局部变量，它的作用域为函数定义范围内。
* 3、函数之间存在作用域互不影响。
* 4、函数内访问全局变量需要 **global** 关键字或者使用 **$GLOBALS[index]** 数组

在 php 中函数是有独立的作用域，所以局部变量会覆盖全局变量，即使局部变量中没有全局变量相同的变量，也会被覆盖。如下：

要想在函数中直接使用全局变量可以通过 **global** 关键字声明或者通过 php 中自定义的 **$GLOBALS** 数组获取:

# PHP 循环 - For 循环

循环执行代码块指定的次数，或者当指定的条件为真时循环执行代码块。

## for 循环

for 循环用于您预先知道脚本需要运行的次数的情况。

参数：

* **初始值**：主要是初始化一个变量值，用于设置一个计数器（但可以是任何在循环的开始被执行一次的代码）。
* **条件**：循环执行的限制条件。如果为 TRUE，则循环继续。如果为 FALSE，则循环结束。
* **增量**：主要用于递增计数器（但可以是任何在循环的结束被执行的代码）。

**注释：**上面的**初始值**和**增量**参数可为空，或者有多个表达式（用逗号分隔）。

## foreach 循环

foreach 循环用于遍历数组。

### **语法**

foreach ($array as $value){

要执行代码;}

每进行一次循环，当前数组元素的值就会被赋值给 $value 变量（数组指针会逐一地移动），在进行下一次循环时，您将看到数组中的下一个值。

# PHP 函数

PHP 的真正威力源自于它的函数。

在 PHP 中，提供了超过 1000 个内建的函数。

## PHP 内建函数

如需查看所有数组函数的完整参考手册和实例，请访问我们的 [PHP 参考手册](https://www.runoob.com/php/php-ref-array.html)。

## PHP 函数

在本章中，我们将为您讲解如何创建自己的函数。

如要在页面加载时执行脚本，您可以把它放到函数里。

函数是通过调用函数来执行的。

你可以在页面的任何位置调用函数。

## 创建 PHP 函数

函数是通过调用函数来执行的

## 创建 PHP 函数

函数是通过调用函数来执行的。

## 语法

<?php function functionName() { // 要执行的代码 } ?>

PHP 函数准则：

* 函数的名称应该提示出它的功能
* 函数名称以字母或下划线开头（不能以数字开头）

## PHP 函数 - 添加参数

为了给函数添加更多的功能，我们可以添加参数，参数类似变量。

参数就在函数名称后面的一个括号内指定。

## PHP 函数 - 返回值

如需让函数返回一个值，请使用 return 语句。

# PHP 魔术常量

PHP 向它运行的任何脚本提供了大量的预定义常量。

不过很多常量都是由不同的扩展库定义的，只有在加载了这些扩展库时才会出现，或者动态加载后，或者在编译时已经包括进去了。

有八个魔术常量它们的值随着它们在代码中的位置改变而改变。

例如 \_\_LINE\_\_ 的值就依赖于它在脚本中所处的行来决定。这些特殊的常量不区分大小写，如下：

## \_\_LINE\_\_

文件中的当前行号。

## 实例

<?php echo '这是第 " ' . \_\_LINE\_\_ . ' " 行'; ?>

## \_\_FILE\_\_

文件的完整路径和文件名。如果用在被包含文件中，则返回被包含的文件名。

自 PHP 4.0.2 起，\_\_FILE\_\_ 总是包含一个绝对路径（如果是符号连接，则是解析后的绝对路径），而在此之前的版本有时会包含一个相对路径。

## \_\_DIR\_\_

文件所在的目录。如果用在被包括文件中，则返回被包括的文件所在的目录。

它等价于 dirname(\_\_FILE\_\_)。除非是根目录，否则目录中名不包括末尾的斜杠。（PHP 5.3.0中新增）

## 实例

<?php echo '该文件位于 " ' . \_\_DIR\_\_ . ' " '; ?>

## \_\_FUNCTION\_\_

函数名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该函数被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

## \_\_CLASS\_\_

类的名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该类被定义时的名字（区分大小写）。

在 PHP 4 中该值总是小写字母的。类名包括其被声明的作用区域（例如 Foo\Bar）。注意自 PHP 5.4 起 \_\_CLASS\_\_ 对 trait 也起作用。当用在 trait 方法中时，\_\_CLASS\_\_ 是调用 trait 方法的类的名字。

## \_\_TRAIT\_\_

Trait 的名字（PHP 5.4.0 新加）。自 PHP 5.4.0 起，PHP 实现了代码复用的一个方法，称为 traits。

Trait 名包括其被声明的作用区域（例如 Foo\Bar）。

从基类继承的成员被插入的 SayWorld Trait 中的 MyHelloWorld 方法所覆盖。其行为 MyHelloWorld 类中定义的方法一致。优先顺序是当前类中的方法会覆盖 trait 方法，而 trait 方法又覆盖了基类中的方法。

## \_\_METHOD\_\_

类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）

## \_\_NAMESPACE\_\_

当前命名空间的名称（区分大小写）。此常量是在编译时定义的（PHP 5.3.0 新增）。

**多个 trait 的情况：**

通过逗号分隔，在 use 声明列出多个 trait，可以都插入到一个类中。示例代码如下：