第三章 变量及数据类型

**世界上唯一不变的变化。**

现代编程语言中的基本元素主要有：变量，流程控制接口，函数等等。 我们能否不使用变量来编写程序呢？这显然是可以的，例如：

<?php

[echo](http://www.php.net/echo) "Hello TIPI Readers";

这个程序很简单，输出一个字符串内容。

就和我们仅仅使用二进制也能编程一样，不使用变量也能完成某些工作， 不使用变量我们的程序将会非常难以编写和维护，变量可以让我们将值存储起来， 以便在程序的其他地方使用，或者通过计算保存新的值。

变量具有三个基本组成部分：

1. **名称** 变量的标示符。就像小狗一样，主人可能会给这些小狗起个喜欢的名称。 变量命名上，PHP继承了Perl的语法风格，变量以美元符号开始，后面跟变量名。 一个有效的变量名由字母或者下划线开头，后面跟上任意数量的字母，数字，或者下划线。 PHP同时还支持复合变量，也就是类似$$a的变量，它会进行两次的解释。这给PHP带来了非常灵活的动态特性。
2. **类型** 变量的类型，就像小狗的品种，不同的小狗血统可能会不一样，有的聪明， 有的会购物等等。 在很多静态语言中，变量在定义时就指定了，在程序运行过程中都不允许进行变更， 那如果你有一只能随便指定品种的小狗会不会很拉风呢;-) PHP就是这样，属于弱类型语言，可以随便赋予它任何类型的值。
3. **值内容**。 这是标示所代表的具体内容。这就像是实实在在的小狗的这个实物。 你可以给任何一条小狗起名为：小七，在编程语言中也是如此，你可以给变量赋予它 所能表示范围的值。不过在同一时间，变量只能有一个值。

PHP中组成变量名的字母可以是英文字母 a-z，A-Z，还可以是 ASCII 字符从 127 到 255（0x7f-0xff）。 变量名是区分大小写的。

除了变量本身，在PHP中我们经常会接触到与变量相关的一些概念，比如：常量，全局变量，静态变量以及类型转换等。 本章我们将介绍这些与变量相关的实现。其中包括PHP本身的变量底层存储结构、弱类型系统的实现以及这些类型之间的相互转换等。

先看一段PHP代码：

<?php

$foo = 10;

$bar = 20;

function change() {

[global](http://www.php.net/global) $foo;

$bar = 0;

$foo++;

}

change();

[echo](http://www.php.net/echo) $foo, ' ', $bar;

运行代码会输出**11 20**。可是为什么会有这样的输出呢？变量在PHP的内部是如何实现的呢？ 变量的作用域又是怎么实现的呢？ 这是本章将对围绕**变量**这个主题展开讨论，下面我们从最基本的变量实现开始。

不是所有编程语言中的变量的值都可以改变的。想想我们学过的数学中的变量。 他们的值也是不可改变的。例如： x + y = 10; 变量x和y的值是不能发生变化的。 在某个具体场景，也就是某个方程式中只有表示特定的值，变量的值不能改变的好处是： 这样就能尽可能少的产生[副作用](http://en.wikipedia.org/wiki/Side_effect_(computer_science))， 在[Erlang语言](http://www.erlang.org/)中就是如此，它是一门函数式编程语言，非常值得学习。

**数据类型**

从类型的维度来看，编程语言可以分为三大类：

1. 静态类型语言，比如：C/Java等，在静态语言类型中，类型的检查是在编译期(compile-time)确定的， 也就是说在运行时变量的类型是不会发生变化的。
2. 动态语言类型，比如：PHP，python等各种脚本语言，这类语言中的类型是在运行时确定的， 那么也就是说类型通常可以在运行时发生变化
3. 无类型语言，比如：汇编语言，汇编语言操作的是底层存储，他们对类型毫无感知。

[« 第四节 小结](http://www.php-internals.com/book/?p=chapt02/02-04-summary)

[第一节 变量的结构和类型 »](http://www.php-internals.com/book/?p=chapt03/03-01-00-variables-structure)