

PHP小白极速入门 —— 变量

主讲人与课程设计：耕耕

变量

PHP 中的变量用一个**美元符号 (\$)** 后面跟变量名来表示。**变量名是区分大小写的。**

变量一般\$开始，然后字母a-z的大小写或者下划线_开头

变量，常量，php.ini
区分大小写，其他不分

```
$var = 'Bob';  
$Var = 'Joe';
```

```
$4site = 'not yet'; // 非法变量名；以数字开头  
$_4site = 'not yet'; // 合法变量名；以下划线开头
```

\$this 是一个特殊的变量，
它不能被赋值

赋值

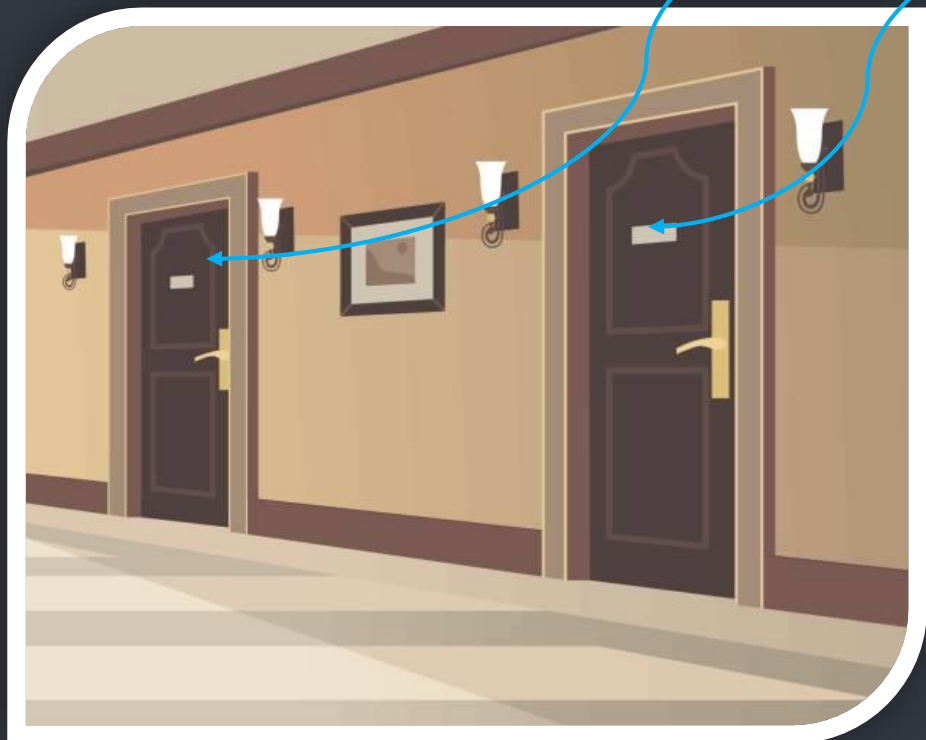
传值赋值（默认），使用“=”将赋值表达式的值赋给另一个变量（**单独分配房间**），相互不干扰

```
$name = "tom";  
$myname = $name;
```

引用赋值：将赋值表达式内存空间的引用赋给另外一个变量，简单地将一个 **&** 符号加到将要赋值的变量前，**共享一个房间**，改变其中一个变量的值会影响另外一个变量

```
$foo = 'Bob'; // 将 'Bob' 赋给 $foo  
$bar = &$foo; // 通过 $bar 引用 $foo
```

传统赋值 (图解)



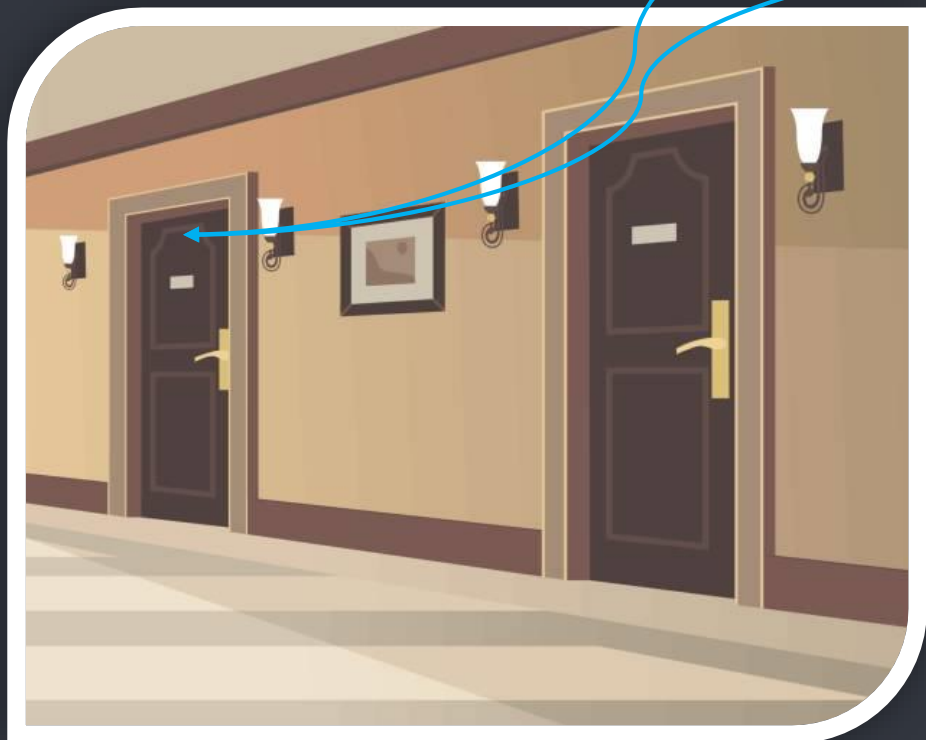
```
$zhangwuji = "单人床";  
$zhaomin = $zhangwuji;
```

```
echo "<h1>$zhangwuji</h1>"; //单人床  
echo "<h1>$zhaomin</h1>"; //单人床  
echo "<h1>-----</h1>";
```

```
$zhangwuji = "双人床";  
echo "<h1>$zhangwuji</h1>"; // 双人床  
echo "<h1>$zhaomin</h1>"; //单人床
```

张无忌和赵敏分别住了两个不同的房间，互不干扰，两把房间钥匙

引用赋值 (图解)



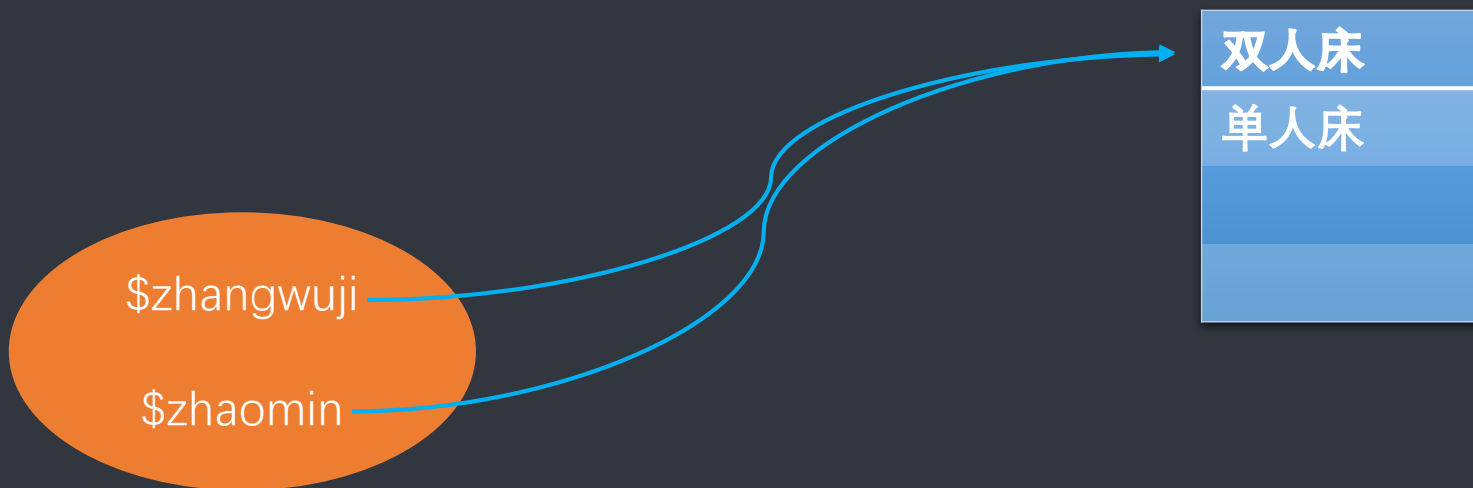
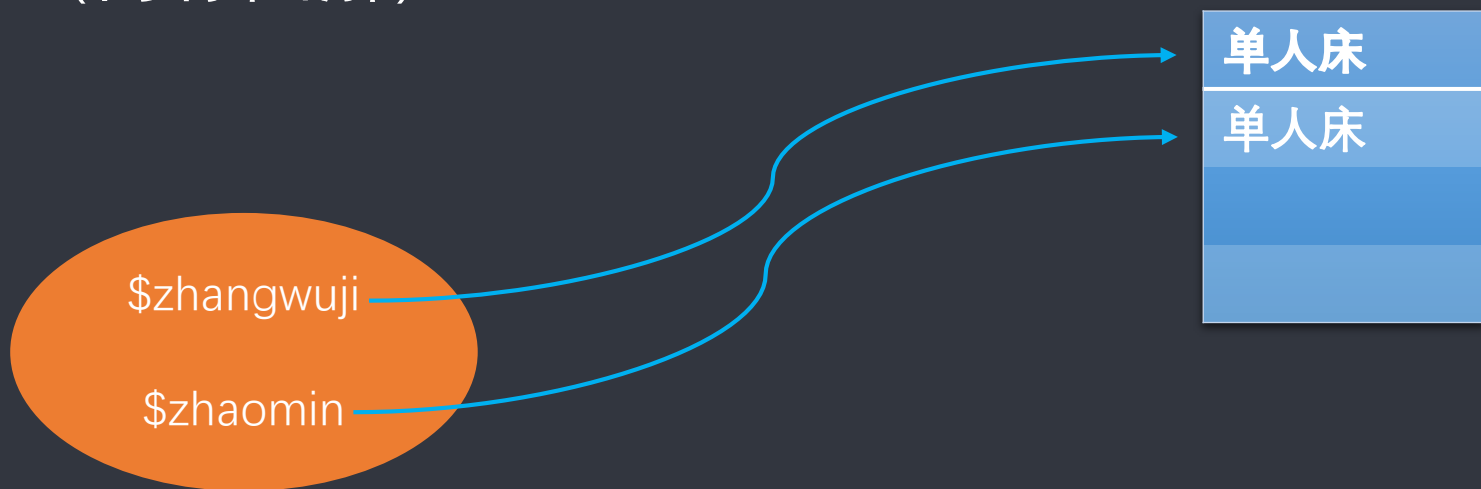
```
$zhangwuji = "单人床";  
$zhaomin = &$zhangwuji;
```

```
echo "<h1>$zhangwuji</h1>"; //单人床  
echo "<h1>$zhaomin</h1>"; //单人床  
echo "<h1>-----</h1>";
```

```
$zhangwuji = "双人床";  
echo "<h1>$zhangwuji</h1>"; // 双人床  
echo "<h1>$zhaomin</h1>"; //双人床
```

张无忌和赵敏住了同一个房间，相互影响，同一把钥匙

赋值 (内存图解)



可变变量

变量的名字可以动态改变，用两个美元(\$\$)符合标示

```
$a = 'hello';  
$$a = 'world'; // 等价于 $hello = 'world'
```

常量

常量是一个简单值的标识符（名字）。如同其名称所暗示的，在脚本执行期间该值**不能改变**

常量默认为大小写敏感。传统上常量标识符总是**大写的**。用**define() 函数**关键词在定义。

常量只能包含标量数据（**boolean, integer, float 和 string**），注意没有**\$**符号

```
define("NAME", "Tom");
```

```
echo NAME;
```

注意：和使用 `define()` 来定义常量相反的是，使用 `const` 关键字定义常量必须处于**最顶端**的作用区域，因为用此方法是在编译时定义的。这就意味着**不能在函数内，循环内以及 if 语句之内**用 `const` 来定义常量。

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**

几个 PHP 的“魔术常量”

名称	说明
<code>__LINE__</code>	文件中的当前行号。
<code>__FILE__</code>	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， <code>__FILE__</code> 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。
<code>__DIR__</code>	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 <code>dirname(__FILE__)</code> 。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） =
<code>__FUNCTION__</code>	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。
<code>__CLASS__</code>	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 <code>Foo\Bar</code> ）。注意自 PHP 5.4 起 <code>__CLASS__</code> 对 trait 也起作用。当用在 trait 方法中时， <code>__CLASS__</code> 是调用 trait 方法的类的名字。
<code>__TRAIT__</code>	Trait 的名字（PHP 5.4.0 新加）。自 PHP 5.4 起此常量返回 trait 被定义时的名字（区分大小写）。Trait 名包括其被声明的作用区域（例如 <code>Foo\Bar</code> ）。
<code>__METHOD__</code>	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）。
<code>__NAMESPACE__</code>	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）。