**PHP认识名词**

**一、变量**

变量是储存信息的容器。

正如代数，PHP 变量可用于保存值（x=5）和表达式（z=x+y）。

变量的名称可以很短（比如 x 和 y），也可以取更具描述性的名称（比如 carname、total\_volume）。

### 1、PHP 变量规则：

* 变量以 $ 符号开头，其后是变量的名称
* 变量名称必须以字母或下划线开头
* 变量名称不能以数字开头
* 变量名称只能包含字母数字字符和下划线（A-z、0-9 以及 \_）
* 变量名称对大小写敏感（$y 与 $Y 是两个不同的变量）

*注释：PHP 变量名称对大小写敏感！*

*注释：如果您为变量赋的值是文本，请用引号包围该值。*

**2、变量的作用域**

变量的作用域指的是变量能够被引用/使用的那部分脚本。

PHP 有三种不同的变量作用域：

* local（局部）
* global（全局）
* static（静态）

## 3、Local 和 Global 作用域

函数之外声明的变量拥有 Global 作用域，只能在函数以外进行访问。

函数内部声明的变量拥有 LOCAL 作用域，只能在函数内部进行访问。

全局变量：在函数外声明

局部变量：函数内声明，且只能在函数内被识别

Global关键词用于访问函数内的全局变量

global $x,$y;

$y=$x+$y;

与下面相同

$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];

## 4、PHP static 关键词

通常，当函数完成/执行后，会删除所有变量。不过，有时我需要不删除某个局部变量。实现这一点需要更进一步的工作。要完成这一点，请在您首次声明变量时使用 **static** 关键词

然后，每当函数被调用时，这个变量所存储的信息都是函数最后一次被调用时所包含的信息。

***注释：****该变量仍然是函数的局部变量。*

<?php

function myTest() {

static $x=0;

echo $x;

$x++;

}

myTest();

myTest();

myTest();

?>

1. **运算符**
2. **算数运算符**

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **名称** | **例子** | **结果** |
| + | 加法 | $x + $y | $x 与 $y 求和 |
| - | 减法 | $x - $y | $x 与 $y 的差数 |
| \* | 乘法 | $x \* $y | $x 与 $y 的乘积 |
| / | 除法 | $x / $y | $x 与 $y 的商数 |
| % | 模数 | $x % $y | $x 除 $y 的余数 |

**2、php赋值运算符**

PHP 赋值运算符用于向变量写值。

PHP 中基础的赋值运算符是 "="。这意味着右侧赋值表达式会为左侧运算数设置值。

|  |  |  |
| --- | --- | --- |
| **赋值** | **等同于** | **描述** |
| x = y | x = y | 右侧表达式为左侧运算数设置值。 |
| x += y | x = x + y | 加 |
| x -= y | x = x - y | 减 |
| x \*= y | x = x \* y | 乘 |
| x /= y | x = x / y | 除 |
| x %= y | x = x % y | 模数 |

## 3、PHP 字符串运算符

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **名称** | **例子** | **结果** |
| . | 串接 | $txt1 = "Hello" $txt2 = $txt1 . " world!" | 现在 $txt2 包含 "Hello world!" |
| .= | 串接赋值 | $txt1 = "Hello" $txt1 .= " world!" | 现在 $txt1 包含 "Hello world!" |

1. **递增递减运算符**

|  |  |  |
| --- | --- | --- |
| **运算符** | **名称** | **描述** |
| ++$x | 前递增 | $x 加一递增，然后返回 $x |
| $x++ | 后递增 | 返回 $x，然后 $x 加一递增 |
| --$x | 前递减 | $x 减一递减，然后返回 $x |
| $x-- | 后递减 | 返回 $x，然后 $x 减一递减 |

## 5、PHP 比较运算符

PHP 比较运算符用于比较两个值（数字或字符串）：

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **名称** | **例子** | **结果** |
| == | 等于 | $x == $y | 如果 $x 等于 $y，则返回 true。 |
| === | 全等（完全相同） | $x === $y | 如果 $x 等于 $y，且它们类型相同，则返回 true。 |
| != | 不等于 | $x != $y | 如果 $x 不等于 $y，则返回 true。 |
| <> | 不等于 | $x <> $y | 如果 $x 不等于 $y，则返回 true。 |
| !== | 不全等（完全不同） | $x !== $y | 如果 $x 不等于 $y，且它们类型不相同，则返回 true。 |
| > | 大于 | $x > $y | 如果 $x 大于 $y，则返回 true。 |
| < | 大于 | $x < $y | 如果 $x 小于 $y，则返回 true。 |
| >= | 大于或等于 | $x >= $y | 如果 $x 大于或者等于 $y，则返回 true. |
| <= | 小于或等于 | $x <= $y | 如果 $x 小于或者等于 $y，则返回 true。 |

## PHP 逻辑运算符

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **名称** | **例子** | **结果** |
| and | 与 | $x and $y | 如果 $x 和 $y 都为 true，则返回 true。 |
| or | 或 | $x or $y | 如果 $x 和 $y 至少有一个为 true，则返回 true。 |
| xor | 异或 | $x xor $y | 如果 $x 和 $y 有且仅有一个为 true，则返回 true。 |
| && | 与 | $x && $y | 如果 $x 和 $y 都为 true，则返回 true。 |
| || | 或 | $x || $y | 如果 $x 和 $y 至少有一个为 true，则返回 true。 |
| ! | 非 | !$x | 如果 $x 不为 true，则返回 true。 |

## 7、PHP 数组运算符

PHP 数组运算符用于比较数组：

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **名称** | **例子** | **结果** |
| + | 联合 | $x + $y | $x 和 $y 的联合（但不覆盖重复的键） |
| == | 相等 | $x == $y | 如果 $x 和 $y 拥有相同的键/值对，则返回 true。 |
| === | 全等 | $x === $y | 如果 $x 和 $y 拥有相同的键/值对，且顺序相同类型相同，则返回 true。 |
| != | 不相等 | $x != $y | 如果 $x 不等于 $y，则返回 true。 |
| <> | 不相等 | $x <> $y | 如果 $x 不等于 $y，则返回 true。 |
| !== | 不全等 | $x !== $y | 如果 $x 与 $y 完全不同，则返回 true。 |

## 三、PHP 常量

常量是单个值的标识符（名称）。在脚本中无法改变该值。

有效的常量名以字符或下划线开头（常量名称前面没有 $ 符号）。

**注释：**与变量不同，常量贯穿整个脚本是自动全局的。

## 1、设置 PHP 常量

如需设置常量，请使用 define() 函数 - 它使用三个参数：

1. 首个参数定义常量的名称
2. 第二个参数定义常量的值
3. 可选的第三个参数规定常量名是否对大小写敏感。默认是 false（对大小写敏感）添加true（对大小写不敏感）。

eg.define("GREETING", "Welcome to W3School.com.cn!", true);

**四、PHP数组**

数组是特殊的变量，它可以同时保存一个以上的值。

数组能够在单一变量名中存储许多值，并且您能够通过引用下标号来访问某个值。

## 1、在 PHP 中创建数组

在 PHP 中， array() 函数用于创建数组：

array();

在 PHP 中，有三种数组类型：

* 索引数组 - 带有数字索引的数组
* 关联数组 - 带有指定键的数组
* 多维数组 - 包含一个或多个数组的数组

## 2、PHP 索引数组

有两种创建索引数组的方法：

索引是自动分配的（索引从 0 开始）：

$cars=array("Volvo","BMW","SAAB");

或者也可以手动分配索引：

$cars[0]="Volvo";

$cars[1]="BMW";

$cars[2]="SAAB";

### 实例

<?php

$cars=array("Volvo","BMW","SAAB");

echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";

?>

## 3、获得数组的长度 - count() 函数

count() 函数用于返回数组的长度（元素数）：

## 4、遍历索引数组

如需遍历并输出索引数组的所有值，您可以使用 for 循环，就像这样：

### 实例

<?php

$cars=array("Volvo","BMW","SAAB");

$arrlength=count($cars);

for($x=0;$x<$arrlength;$x++) {

echo $cars[$x];

echo "<br>";

}

?>

## 5、PHP 关联数组

关联数组是使用您分配给数组的指定键的数组。

有两种创建关联数组的方法：

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

或者：

$age['Peter']="35";

$age['Ben']="37";

$age['Joe']="43";

随后可以在脚本中使用指定键：

### 实例

<?php

$age=array("Bill"=>"35","Steve"=>"37","Peter"=>"43");

echo "Peter is " . $age['Peter'] . " years old.";

?>

## 6、遍历关联数组

如需遍历并输出关联数组的所有值，您可以使用 foreach 循环，就像这样：

### 实例

<?php

$age=array("Bill"=>"35","Steve"=>"37","Peter"=>"43");

foreach($age as $x=>$x\_value) {

echo "Key=" . $x . ", Value=" . $x\_value;

echo "<br>";

}

?>

## 7、多维数组

## 五、PHP 循环

在您编写代码时，经常需要反复运行同一代码块。我们可以使用循环来执行这样的任务，而不是在脚本中添加若干几乎相等的代码行。

在 PHP 中，我们有以下循环语句：

* **while** - 只要指定条件为真，则循环代码块
* **do...while** - 先执行一次代码块，然后只要指定条件为真则重复循环
* **for** - 循环代码块指定次数
* **foreach** - 遍历数组中的每个元素并循环代码块

## 1、PHP while 循环

只要指定的条件为真，while 循环就会执行代码块。

### 语法

while (条件为真) {

要执行的代码;

}

## 2、PHP do...while 循环

do...while 循环首先会执行一次代码块，然后检查条件，如果指定条件为真，则重复循环。

### 语法

do {

要执行的代码;

} while (条件为真);

### 实例

<?php

$x=1;

do {

echo "这个数字是：$x <br>";

$x++;

} while ($x<=5);

?>

## 3、PHP for 循环

如果您已经提前确定脚本运行的次数，可以使用 for 循环。

### 语法

for (init counter; test counter; increment counter) {

code to be executed;

}

**参数**：

* init counter：初始化循环计数器的值
* test counter：: 评估每个循环迭代。如果值为 TRUE，继续循环。如果它的值为 FALSE，循环结束。
* increment counter：增加循环计数器的值

下面的例子显示了从 0 到 10 的数字：

### 实例

<?php

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

echo "数字是：$x <br>";

}

?>

## 4、PHP foreach 循环

foreach 循环只适用于数组，并用于遍历数组中的每个键/值对。

### 语法

foreach ($array as $value) {

code to be executed;

}

每进行一次循环迭代，当前数组元素的值就会被赋值给 $value 变量，并且数组指针会逐一地移动，直到到达最后一个数组元素。

下面的例子演示的循环将输出给定数组（$colors）的值：

### 实例

<?php

$colors = array("red","green","blue","yellow");

foreach ($colors as $value) {

echo "$value <br>";

}

?>

# PHP 函数

## 1、PHP 用户定义函数

除了内建的 PHP 函数，我们可以创建我们自己的函数。

函数是可以在程序中重复使用的语句块。

页面加载时函数不会立即执行。

函数只有在被调用时才会执行。

## 2、在 PHP 创建用户定义函数

用户定义的函数声明以关单 "function" 开头：

### 语法

function functionName() {

被执行的代码;

}

注释：函数名能够以字母或下划线开头（而非数字）。

注释：函数名对大小写不敏感。

提示：函数名应该能够反映函数所执行的任务。

在下面的例子中，我们创建名为 "writeMsg()" 的函数。打开的花括号（{）指示函数代码的开始，而关闭的花括号（}）指示函数的结束。此函数输出 "Hello world!"。如需调用该函数，只要使用函数名即可：

### 实例

<?php

function writeMsg() {

echo "Hello world!";

}

writeMsg(); // 调用函数

?>

## 3、PHP 函数参数

可以通过参数向函数传递信息。参数类似变量。

参数被定义在函数名之后，括号内部。您可以添加任意多参数，只要用逗号隔开即可。

下面的例子中的函数有一个参数（$fname）。当调用 familyName() 函数时，我们同时要传递一个名字（例如 Bill），这样会输出不同的名字，但是姓氏相同：

### 实例

<?php

function familyName($fname) {

echo "$fname Zhang.<br>";

}

familyName("Li");

familyName("Hong");

familyName("Tao");

familyName("Xiao Mei");

familyName("Jian");

?>

## 4、PHP 默认参数值

下面的例子展示了如何使用默认参数。如果我们调用没有参数的 setHeight() 函数，它的参数会取默认值：

### 实例

<?php

function setHeight($minheight=50) {

echo "The height is : $minheight <br>";

}

setHeight(350);

setHeight(); // 将使用默认值 50

setHeight(135);

setHeight(80);

?>

## 5、PHP 函数 - 返回值

如需使函数返回值，请使用 return 语句：

## **Cookie**（储存在用户本地终端上的数据）

Cookie，有时也用其复数形式 [Cookies](http://baike.baidu.com/subview/1311/6319629.htm" \t "http://baike.baidu.com/_blank)，指某些网站为了辨别用户身份、进行 session 跟踪而储存在用户本地终端上的数据（通常经过加密）。定义于 RFC2109 和 2965 中的都已废弃，最新取代的规范是 RFC6265[1]  。（可以叫做浏览器缓存）

# 八、session（计算机语言）

Session:在计算机中，尤其是在网络应用中，称为“会话控制”。Session 对象存储特定用户会话所需的属性及配置信息。这样，当用户在应用程序的 Web 页之间跳转时，存储在 Session 对象中的变量将不会丢失，而是在整个用户会话中一直存在下去。当用户请求来自应用程序的 Web 页时，如果该用户还没有会话，则 Web 服务器将自动创建一个 Session 对象。当会话过期或被放弃后，服务器将终止该会话。Session 对象最常见的一个用法就是存储用户的首选项。例如，如果用户指明不喜欢查看图形，就可以将该信息存储在 Session 对象中。有关使用 Session 对象的详细信息，请参阅“ASP 应用程序”部分的“管理会话”。注意 会话状态仅在支持 cookie 的浏览器中保留。

# 九、mySQL（关系型数据库管理系统）

MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](http://baike.baidu.com/view/1088.htm" \t "http://baike.baidu.com/_blank)的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](http://baike.baidu.com/view/394804.htm" \t "http://baike.baidu.com/_blank)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

由于其社区版的性能卓越，搭配 [PHP](http://baike.baidu.com/view/99.htm" \t "http://baike.baidu.com/_blank) 和 [Apache](http://baike.baidu.com/view/28283.htm" \t "http://baike.baidu.com/_blank) 可组成良好的开发环境。

1. **Mysqli**

mysqli 扩展允许用户访问由 MySQL 4.1 或更高版本所提供的功能。作用：文档中包括MySQL 手册的内容。协同工作。

1. **AJAX（AJAX开发）**

AJAX即“**A***synchronous* **J***avascript* ***A****nd* **X***ML*”（异步JavaScript和XML），是指一种创建交互式[网页](http://baike.baidu.com/view/828.htm" \t "http://baike.baidu.com/_blank)应用的网页开发技术。

AJAX = 异步 [JavaScript](http://baike.baidu.com/view/16168.htm" \t "http://baike.baidu.com/_blank)和[XML](http://baike.baidu.com/view/63.htm" \t "http://baike.baidu.com/_blank)（[标准通用标记语言](http://baike.baidu.com/view/5286041.htm" \t "http://baike.baidu.com/_blank)的子集）。

AJAX 是一种用于创建快速动态网页的技术。

通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

传统的网页（不使用 AJAX）如果需要更新内容，必须重载整个网页页面。

# 十二、域名

[域名](http://baike.baidu.com/view/43.htm" \t "http://baike.baidu.com/item/_blank)（[Domain Name](http://baike.baidu.com/view/105602.htm" \t "http://baike.baidu.com/item/_blank)），是由一串用点分隔的名字组成的Internet上某一台计算机或计算机组的名称，用于在数据传输时标识计算机的电子方位（有时也指地理位置，地理上的域名，指代有行政自主权的一个地方区域）。域名是一个IP地址上有“面具” 。一个域名的目的是便于记忆和沟通的一组服务器的地址（网站，电子邮件，[FTP](http://baike.baidu.com/subview/369/6149695.htm" \t "http://baike.baidu.com/item/_blank)等）。域名作为力所能及难忘的互联网参与者的名称，世界上第一个注册的域名是在1985年1月注册的。

1. **url**

统一[资源](http://baike.baidu.com/view/8439.htm" \t "http://baike.baidu.com/item/_blank)定位符是对可以从[互联网](http://baike.baidu.com/view/6825.htm" \t "http://baike.baidu.com/item/_blank)上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它。[1]

它最初是由[蒂姆·伯纳斯·李](http://baike.baidu.com/view/1232221.htm" \t "http://baike.baidu.com/item/_blank)发明用来作为[万维网](http://baike.baidu.com/view/7833.htm" \t "http://baike.baidu.com/item/_blank)的地址。现在它已经被万维网联盟编制为互联网标准RFC1738了

**十四、服务器**

服务器，也称伺服器，是提供计算服务的设备。由于服务器需要响应服务请求，并进行处理，因此一般来说服务器应具备承担服务并且保障服务的能力。

服务器的构成包括[处理器](http://baike.baidu.com/view/50152.htm" \t "http://baike.baidu.com/item/_blank)、[硬盘](http://baike.baidu.com/view/4480.htm" \t "http://baike.baidu.com/item/_blank)、[内存](http://baike.baidu.com/view/1082.htm" \t "http://baike.baidu.com/item/_blank)、[系统](http://baike.baidu.com/view/25302.htm" \t "http://baike.baidu.com/item/_blank)[总线](http://baike.baidu.com/view/1389.htm" \t "http://baike.baidu.com/item/_blank)等，和通用的计算机架构类似，但是由于需要[提供](http://baike.baidu.com/view/834302.htm" \t "http://baike.baidu.com/item/_blank)高可靠的服务，因此在处理能力、稳定性、可靠性、安全性、可扩展性、可管理性等方面要求较高。

在网络环境下，根据服务器提供的服务类型不同，分为文件服务器，[数据库服务器](http://baike.baidu.com/view/32944.htm" \t "http://baike.baidu.com/item/_blank)，应用程序服务器，WEB服务器等。

**十五、linux**

Linux是一套免费使用和自由传播的[类Unix](http://baike.baidu.com/view/3289073.htm" \t "http://baike.baidu.com/item/_blank)[操作系统](http://baike.baidu.com/subview/880/4940471.htm" \t "http://baike.baidu.com/item/_blank)，是一个基于[POSIX](http://baike.baidu.com/view/209573.htm" \t "http://baike.baidu.com/item/_blank)和[UNIX](http://baike.baidu.com/view/8095.htm" \t "http://baike.baidu.com/item/_blank)的多用户、[多任务](http://baike.baidu.com/view/757238.htm" \t "http://baike.baidu.com/item/_blank)、支持[多线程](http://baike.baidu.com/view/65706.htm" \t "http://baike.baidu.com/item/_blank)和多[CPU](http://baike.baidu.com/view/2089.htm" \t "http://baike.baidu.com/item/_blank)的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持[32位](http://baike.baidu.com/subview/125389/10889278.htm" \t "http://baike.baidu.com/item/_blank)和[64位](http://baike.baidu.com/view/125381.htm" \t "http://baike.baidu.com/item/_blank)硬件。Linux继承了[Unix](http://baike.baidu.com/view/8095.htm" \t "http://baike.baidu.com/item/_blank)以[网络](http://baike.baidu.com/view/3487.htm" \t "http://baike.baidu.com/item/_blank)为核心的设计思想，是一个性能稳定的多用户网络操作系统。

**十六、虚拟机**

虚拟机（Virtual Machine）指通过[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/item/_blank)模拟的具有完整[硬件](http://baike.baidu.com/view/25278.htm" \t "http://baike.baidu.com/item/_blank)系统功能的、运行在一个完全[隔离](http://baike.baidu.com/view/292853.htm" \t "http://baike.baidu.com/item/_blank)环境中的完整[计算机系统](http://baike.baidu.com/view/1130583.htm" \t "http://baike.baidu.com/item/_blank)。

**十七、ubuntu**

Ubuntu（乌班图）是一个以[桌面](http://baike.baidu.com/view/79807.htm" \t "http://baike.baidu.com/item/_blank)应用为主的[Linux操作系统](http://baike.baidu.com/view/46577.htm" \t "http://baike.baidu.com/item/_blank)。

**十八、centos**

CentOS（Community Enterprise Operating System，中文意思是：社区企业操作系统）是Linux发行版之一，它是来自于Red Hat Enterprise Linux依照[开放源代码](http://baike.baidu.com/view/1708.htm" \t "http://baike.baidu.com/item/_blank)规定释出的源代码所编译而成。由于出自同样的[源代码](http://baike.baidu.com/view/60376.htm" \t "http://baike.baidu.com/item/_blank)，因此有些要求高度稳定性的[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/item/_blank)以CentOS替代商业版的[Red Hat](http://baike.baidu.com/view/1139590.htm" \t "http://baike.baidu.com/item/_blank) Enterprise Linux使用。两者的不同，在于CentOS并不包含封闭源代码软件。

**十九、apache**

Apache是世界使用排名第一的Web[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/item/apache/_blank)软件。它可以运行在几乎所有广泛使用的[计算机平台](http://baike.baidu.com/view/2269685.htm" \t "http://baike.baidu.com/item/apache/_blank)上，由于其[跨平台](http://baike.baidu.com/view/469855.htm" \t "http://baike.baidu.com/item/apache/_blank)和安全性被广泛使用，是最流行的Web服务器端软件之一。它快速、可靠并且可通过简单的API扩充，将[Perl](http://baike.baidu.com/view/46614.htm" \t "http://baike.baidu.com/item/apache/_blank)/[Python](http://baike.baidu.com/view/21087.htm" \t "http://baike.baidu.com/item/apache/_blank)等[解释器](http://baike.baidu.com/view/592974.htm" \t "http://baike.baidu.com/item/apache/_blank)编译到服务器中。

**二十、nginx**

***Nginx*** ("engine x") 是一个高性能的[HTTP](http://baike.baidu.com/view/9472.htm" \t "http://baike.baidu.com/item/_blank)和[反向代理](http://baike.baidu.com/view/1165595.htm" \t "http://baike.baidu.com/item/_blank)服务器，也是一个IMAP/POP3/SMTP[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/item/_blank)。**Nginx**是一款[轻量级](http://baike.baidu.com/subview/1318763/16205192.htm" \t "http://baike.baidu.com/item/_blank)的[Web](http://baike.baidu.com/subview/3912/15992867.htm" \t "http://baike.baidu.com/item/_blank) 服务器/[反向代理](http://baike.baidu.com/view/1165595.htm" \t "http://baike.baidu.com/item/_blank)服务器及[电子邮件](http://baike.baidu.com/view/1524.htm" \t "http://baike.baidu.com/item/_blank)（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。

## **二十一、Git**（分布式版本控制系统）

Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。Git是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

# **二十二、**SVN

# SVN是Subversion的简称，是一个开放源代码的版本控制系统，相较于RCS、CVS，它采用了分支管理系统，它的设计目标就是取代CVS。互联网上很多版本控制服务已从CVS迁移到Subversion。说得简单一点SVN就是用于多个人共同开发同一个项目，共用资源的目的。