**PHP生成随机字符串（3种方法）**

如用户注册生成随机密码，用户重置密码也需要生成一个随机的密码。随机密码也就是一串固定长度的字符串，文章整理了几种生成随机字符串的方法。  
**方法一**

1、在33 – 126中生成一个随机整数，如35。

2、将35转换成对应的ASCII码字符，如35对应#。

3、重复以上1、2步骤n次，连接成n位的密码。

该算法主要用到了两个函数，mt\_rand ( int $min , int $max )函数用于生成随机整数，其中 $min – $max 为 ASCII 码的范围，这里取 33 -126 ，可以根据需要调整范围，如ASCII码表中 97 – 122 位对应 a – z 的英文字母，具体可参考 ASCII码表； chr ( int $ascii )函数用于将对应整数 $ascii 转换成对应的字符。

[?](http://www.jb51.net/article/72758.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | function create\_password($pw\_length )  {  $randpwd = ”;  for ($i = 0; $i **<** $pw\_length; $i++)  {  $randpwd .= chr(mt\_rand(33, 126));  }  return $randpwd;  }  // 调用该函数，传递长度参数$pw\_length = 6  echo create\_password(6); |

**方法二**

1、预置一个的字符串 $chars ，包括 a – z，A – Z，0 – 9，以及一些特殊字符。

2、在 $chars 字符串中随机取一个字符。

3、重复第二步n次，可得长度为n的密码。

[?](http://www.jb51.net/article/72758.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | function generate\_password( $length = 8 ) {  // 密码字符集，可任意添加你需要的字符  $chars = ‘abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&\*()-\_ []{}**<>**~`+=,.;:/?|’;  $password = ”;  for ( $i = 0; $i **<** $length; $i++ )  {  // 这里提供两种字符获取方式  // 第一种是使用 substr 截取$chars中的任意一位字符；  // 第二种是取字符数组 $chars 的任意元素  // $password .= substr($chars, mt\_rand(0, strlen($chars) – 1), 1);  $password .= $chars[ mt\_rand(0, strlen($chars) - 1) ];  }  return $password;  } |

**方法三**

1、预置一个的字符数组 $chars ，包括 a – z，A – Z，0 – 9，以及一些特殊字符。

2、通过array\_rand()从数组$chars中随机选出$length个元素。

3、根据已获取的键名数组 $keys，从数组$chars取出字符拼接字符串。该方法的缺点是相同的字符不会重复取。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | function make\_password( $length = 8 )  {    // 密码字符集，可任意添加你需要的字符  $chars = array(‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’,  ‘i’, ‘j’, ‘k’, ‘l’,'m’, ‘n’, ‘o’, ‘p’, ‘q’, ‘r’, ’s’,  ‘t’, ‘u’, ‘v’, ‘w’, ‘x’, ‘y’,'z’, ‘A’, ‘B’, ‘C’, ‘D’,  ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘J’, ‘K’, ‘L’,'M’, ‘N’, ‘O’,  ‘P’, ‘Q’, ‘R’, ‘S’, ‘T’, ‘U’, ‘V’, ‘W’, ‘X’, ‘Y’,'Z’,  ‘0′, ‘1′, ‘2′, ‘3′, ‘4′, ‘5′, ‘6′, ‘7′, ‘8′, ‘9′, ‘!’,  ‘@’,'#’, ‘$’, ‘%’, ‘^’, ‘&’, ‘\*’, ‘(‘, ‘)’, ‘-’, ‘\_’,  ‘[', ']‘, ‘{‘, ‘}’, ‘**<**’, ‘**>**’, ‘~’, ‘`’, ‘+’, ‘=’, ‘,’,  ‘.’, ‘;’, ‘:’, ‘/’, ‘?’, ‘|’);    // 在 $chars 中随机取 $length 个数组元素键名  $keys = ($chars, $length);  $password = ”;  for($i = 0; $i **<** $length; $i++)  {  // 将 $length 个数组元素连接成字符串  $password .= $chars[$keys[$i]];  }  return $password;  } |

**时间效率对比**

我们使用以下PHP代码，计算上面的3 个随机密码生成函数生成6 位密码的运行时间，进而对他们的时间效率进行一个简单的对比。最终得出的结果是：

**方法一：9.8943710327148E-5 秒    
方法二：9.6797943115234E-5 秒    
方法三：0.00017499923706055 秒**可以看出方法一和方法二的执行时间都差不多，而方法三的运行时间稍微长了点。  
通过比较三种方法的实现过程，时间效率对比，学习了PHP生成随机字符串的三种方法，希望可以对大家今后的学习有所帮助。