

[← Perl 进程管理](#)

## Perl POD 文档

Perl 中可以在模块或脚本中嵌入 POD ( Plain Old Documentation ) 文档。

POD 是一种简单而易用的标记型语言 ( 置标语言 )。

POD 文档使用规则：

*POD 文档以 **=head1** 开始，**=cut** 结束，**=head1** 前与**=cut** 后添加一空行。*

Perl 会忽略 POD 中的文档。实例如下：

### 实例

```
#!/usr/bin/perl
print "Hello, World\n";
=head1 Hello, World 实例
这是一个 Perl 的简单实例。
=cut
print "Hello, Runoob\n";
```

执行以上程序，输出结果为：

```
Hello, World
Hello, Runoob
```

我们还可以使用 "\_\_END\_\_" 或 "\_\_DATA\_\_" 将所在行之后的内容全部"注释"掉：

### 实例

```
#!/usr/bin/perl
print "Hello, World\n";
while(<DATA>){
print $_;
}
__END__
=head1 Hello, World 实例
这是一个 Perl 的简单实例。
print "Hello, Runoob\n";
```

执行以上程序，输出结果为：

```
Hello, World

=head1 Hello, World 实例
这是一个 Perl 的简单实例。
print "Hello, Runoob\n";
```

以下实例不读取 POD 文档：

### 实例

```
#!/usr/bin/perl
print "Hello, World\n";
__END__
=head1 Hello, World 实例
这是一个 Perl 的简单实例。
print "Hello, Runoob\n";
```

执行以上程序，输出结果为：

```
Hello, World
```

## 什么是 POD ？

Pod(Plain Old Documentation), 是一种简单而易用的标记型语言（置标语言），它经常用于在perl程序和模块中的文档书写。

Pod 的 转化器可以将 Pod 转换成很多种格式，例如 text, html, man 等很多。

Pod 标记语言包含三种基本基本类型：普通, 原文, 和 命令。

- **普通段落:** 你可以在普通段落中使用格式化代码，如黑体，斜体，或代码风格，下划线等。
- **原文段落:** 原文段落，用于代码块或者其他不需要转换器处理的部分，而且不需要段落重排。
- **命令段落:** 命令段落作用于整个的文档，通常用于标题设置或列表标记。

所有的命令段落（他只有一行的长度）使用 "=" 开始，然后是一个标识符。随后的文本将被这条命令所影响。现在被广泛使用的命令包括

```
=pod (开始文档)
=head1 标题文本
=head2 标题文本
=head3 标题文本
=head4 标题文本
=over 缩进空格数量
=item 前缀
=back (结束列表)
=begin 文档格式
=end 结束文档格式
=for 格式文本
=encoding 编码类型
=cut (文档结束)
```

在perl中，可以使用 `pod2html *.pod >*.html` 来生成html格式的pod文档。

考虑以下 POD 实例：

### 实例

```
=begin html
=encoding utf-8
=head1 菜鸟教程
=cut
```

pod2html时会原文拷贝此段代码。

使用 pod2html 命令执行，将其转换为 HTML 代码：

```
$ pod2html test.pod > test.html
```

在浏览器中打开 test.html，链接部分为索引，显示如下：

- [菜鸟教程](#)

## 菜鸟教程

以下实例在 POD 文档中直接写入 HTML：

```
=begin html
=encoding utf-8

<h1>菜鸟教程</h1>
<p> www.runoob.com </p>

=end html
```

pod2html时会原文拷贝此段代码。

使用 pod2html 命令执行，将其转换为 HTML 代码：

```
$ pod2html test.pod > test.html
```

在浏览器中打开 test.html，链接部分为索引，显示如下：

## 菜鸟教程

www.runoob.com

← Perl 进程管理

 点我分享笔记

