

Perl 内置特殊变量总结

【 文件句柄 】

- \$| 如果非零，则在对当前选定的文件执行写或打印操作后强制清除缓冲区
- \$% 当前选中文件句柄的当前页码
- \$= 当前选中文件句柄的当前页面长度
- \$- 当前选中文件句柄的页面剩余长度
- \$~ 当前选中文件句柄的报表格式名称
- \$^ 当前选中文件句柄的当前页眉格式名

【 全局变量 】

- \$_ 默认的输入和模式搜索空间
- \$@ 上一个 eval、do 或 require 函数产生的错误消息
- \$/ 输入记录的分隔符，默认为换行符（类似于 awk 中的 RS）
- \\ 打印函数的输入记录分隔符。除非设置了 \$\\="\n" 否则不会打印换行符
- \$# 使用打印函数打印的数字输出格式(类似于 awk 中的 OMFT)
- ? 上一个关闭的管道、反引号中的命令以及系统函数的返回状态
- ! 根据上下文内容，返回错误号或者错误串
- \$\$ 运行本脚本的 Perl 程序进程 ID
- * 默认值为 0。如果为 1 的话，表示在字符串中进行多行匹配；0 表示只在单行中匹配
- \$, 打印函数的输出字段分隔符。一般在以逗号隔开的字符串中不打印其定界符，除非设置了 \$,=""
- " 与 \$, 类似，但它是在双引号中提供打印数组的，默认值为空格
- \$0 Perl 脚本名
- [指向数组或子串的第一个字符的索引，默认值为 0
-] 当使用 Perl -v 命令获得版本信息时打印出字符串的第一部分
- ; 多维数组的下标分隔符。默认为 \034 （类似于 awk 的 SUBSEP）
- . 读取上一个文件句柄的当前输入行号；必须在关闭前一个文件句柄后才能对下一个文件句柄重置行号
- < 进程的实际 uid
- > 进程的有效 uid
- (进程的实际 gid
-) 进程的有效 gid
- : 字符集，将其后的字符串断开，并以某种格式填入连续的行（从 ^ 处开始）。默是以空白字符、换行符或冒号上隔开的 \n-
- ^A formline 和 write 操作的累加器
- ^C 如果 Perl 使用 -c 选项在只编译的模式下运行，则返回 TRUE

`$_D` 使用 `-D` 开关打开 Perl 调试标记

`$_E` 操作系统的相关出错信息

`$_F` 传递给子例程的最大文件描述符数，一般是 2

`$_H` 语法检查的当前状态

`$_I` 在使用 `-i` 开关时，原位编辑扩展的当前值

`$_L` 进纸符，用在格式中

`$_M` 紧急内存池

`$_O` 操作系统名

`$_P` 内部 Perl 调试标记

`$_S` Perl 解释器的状态

`$_T` 脚本执行时间。用于 `-A`、`-C` 和 `-M` 测试运算符，可设置为 `time` 返回的任何数字值，执行与当前时间有关的文件测试

`$_V` Perl 版本

`$_W` 警告开关的当前设定情况

`$_X` Perl 调用的全路径名

`_` 下划线，负责在声明文件时测试文件的特殊标识符

`ARGV` 用于行参数循环的特殊文件句柄数组

`$ARGV` 从 `<ARGV>` 读取时，含有当前文件名的变量

`@ARGV` 含有命令行参数的数组

`DATA` 一个特殊文件句柄变量，负责引用 `_END_` 标志后面的内容

`@F` 在使用 `-a` 开关时，自动拆分输入行所得到的数组

`@INC` 一个含有文件名路径的数组，`require` 和 `do` 函数从中搜索将要导入的文件名路径

`%INC` 一个含有文件名路径的关联数组，`require` 和 `do` 函数将从中搜索将要导入的文件名路径。其中键是文件名，值是文件路径位置

`%ENV` 含有当前环境变量的关联数组

`@EXPORT` 导出时的默认符号

`@EXPORT_OK` 根据用户请求而导出的符号

`%EXPORT_TAGS` 用于 `Exporter.pm` 中，负责对符号集合统一命名

`%SIG` 用于设置信号处理例程的关联数组

`STDERR` 用于标准出错内容的特殊文件句柄

`STDIN` 用于标准输入的特殊文件句柄

`STDOUT` 用于标准输出内容的特殊文件句柄

`$_[0]` 和 `$_+[0]` 代表当前匹配的正则表达式在被匹配的字符串中的起始和终止的位置

【 块内局部变量 】

- \$1..\$9 含有与对应的一组括号的子记忆模式。（等效于\1..\9）
- \$& 上一次与模式相匹配的字符串（等效于 sed 编辑器）
- \$` 上一次与模式相匹配的项的前导字符串
- \$' 上一次与模式相匹配的项之后的字符串
- \$+ 与搜索模式相匹配的最后一个模式
- @_ 传给子程序的参数列表

附：另一篇总结

一、正则表达式特殊变量：

- 1、\$n 包含上次模式匹配的第 n 个子串
- 2、\$& 前一次成功模式匹配的字符串
- 3、\$` 前次匹配成功的子串之前的内容
- 4、\$' 前次匹配成功的子串之后的内容
- 5、\$+ 前一次使用括号的模式匹配的字符串。

二、文件句柄特殊变量：

- 1、\$| 如果设置为零,在每次调用函数 write 或 print 后, 自动调用函数 fflush,将所写内容写回文件
- 2、\$% 当前输出页号
- 3、\$= 当前每页长度
- 4、\$- 当前页剩余的行数
- 5、\$~ 当前报表输出格式的名称.默认值是文件句柄名.
- 6、\$^ 当前报表输出表头格式的名称.默认值是带后缀"_TOP"的文件句柄名.

三、全局特殊变量：

- 1、\$_和\$ARG 默认输入和模式匹配内容.如果不指定接收输入的变量或执行模式匹配的字符串
- 2、\$.和\$NR 前一次读的文件句柄的当前行号
- 3、\$/和\$RS 输入记录分隔符,默认是换行字符.如用 undef 这个变量,将读到文件结尾
- 4、\$,的\$OFS 输出域分隔符.
- 5、\$\和\$ORS 输出记录分隔符
- 6、\$: 断行字符,默认是"\n-"三个字符
- 7、\$! 这个变量的数字值是 errno 的值,字符串值是对应的系统错误字符串
- 8、\$@ 命令 eval 的错误消息.如果为空,则表示上一次 eval 命令执行成功
- 9、\$\$ 运行当前 Perl 脚本程序的进程号

- 10、\$< 当前进程的实际用户号
- 11、\$> 当前进程的有效用户号
- 12、\$(当前进程的实际组用户号
- 13、\$) 当前进程的有效组用户号
- 14、\$[数组的数组第一个元素的下标,默认是 0
- 15、\$] Perl 的版本号
- 16、\$ARGV 当使用"< >"读文件时,保存当前文件名

四、全局特殊数组变量:

- 1、@ARGV 保存命令行参数
- 2、@INC 保存查找 Perl 程序的目录
- 3、@_ 在子程序内,这个数组变量包含传递给子程序的参数
- 4、%INC 散列表%INC 包含所有用 do 或 require 语句包含的文件,关键字是文件名,是这个文件的路径
- 5、%ENV 包含当前环境变量

五、五、全局特殊文件句柄

- 1、ARGV 遍历数组变量@ARGV 中的所有文件名的特殊文件句柄
- 2、STDERR 标准错误输出句柄
- 3、STDIN 标准输入句柄
- 4、STDOUT 标准输出句柄