

VARIABILI

Negli script in perl, i dati temporanei sono tenuti in variabili; queste si comportano a grandi linee come le variabili d'ambiente, ma non sono la stessa cosa.

Il tipo principale di variabile, in perl, è lo **scalare**. Uno scalare è una stringa o un numero, che non ha bisogno di essere convertito come in C. Inoltre, tutti gli scalari possono essere interi o numeri in virgola mobile, che perl proverà a rappresentare come interi, se possibile, per migliorare la velocità di esecuzione.

Ogni variabile ha un nome, che inizia sempre con un carattere speciale, che ne indica il tipo, seguito da un nome di almeno una lettera. Tutti i nomi delle variabili sono case-sensitive, per cui \$variabile è diverso da \$Variabile e da \$VaRiAbIlE.

Il carattere speciale di cui si parlava è il simbolo del dollaro (\$), ed è utilizzato sia quando si assegna una variabile, sia quando si usa il suo valore.

Ecco qualche esempio:

Assegna il valore intero 1 alla variabile scalare \$bla

```
$bla = 1;
```

Assegna la stringa "1" a \$bla, che è equivalente a sopra

```
$bla = 1;
```

\$ops vale 2

```
$ops = $bla + 1;
```

Visualizza "ciao" (n è un ritorno a capo)

```
$bla = "ciao";  
print $bla;
```

Negli script di shell, il carattere \$ è utilizzato solo quando si richiede il valore di una variabile, non quando lo si assegna:

```
bla = 1  
echo $bla
```

Negli script, le stringhe sono racchiuse tra apici (") o doppi apici (""), con la differenza che tra doppi apici ("") le variabili verranno interpolate, mentre tra apici singoli (") le variabili verranno considerate letteralmente.

Ogni volta che può nascere un'ambiguità come quanto lungo può essere il nome di una variabile, \$variabile può essere scritto come \${variabile}.

Visualizza "uff"

```
$bla = "eeek!";  
$blabla = "uff!";  
print "$blablan";
```

visualizza "eeek!bla"

```
print "${bla}blan"
```

\$msg contiene "eeek! mi hai spaventato"

```
$bla = "eeek!";  
$msg = "$bla mi hai spaventato";
```

\$msg2 contiene "\$bla mi hai spaventato"

```
$msg2 = '$bla mi hai spaventato';
```

In perl, come anche nello shell scripting (e diversamente dal C) le variabili non hanno bisogno di essere dichiarate prima di essere usate: la prima volta che si utilizza una variabile, la si dichiara. Tentare di leggere un valore inesistente porta ad avere una stringa vuota che, in contesto numerico, è interpretata come 0.

Le variabili il cui nome inizia (dopo il canonico \$) con un carattere speciale, che non è una lettera, sono considerate 'riservate' (una lista la si può trovare digitando man perlvar con perl5). Queste variabili cambiano alcuni comportamenti del perl a seconda del proprio valore. Una di queste è la variabile \$_, che è lasciata libera in modo da permettere all'utente di inserirvi dati in modo temporaneo, ed è utile anche per un altro aspetto: molte funzioni operano con \$_ se non è fornito alcun valore.

visualizza il contenuto di \$_

```
$_ = "bla bla blan";  
print;
```

Non tutte le funzioni assumono `$_` come parametro di default, anche se questo funziona con tutte le funzioni in cui ha senso assumere la variabile `$_`. In caso di dubbio, si può ricorrere ad un piccolo script di un paio di righe per vedere se la funzione che interessa può lavorare con `$_`.

Altre importanti variabili riservate sono:

Variabile riservata	Descrizione
<code>\$\$</code>	visualizza il PID del programma
<code>\$<</code>	visualizza l'UID dell'utente che ha eseguito il programma
<code>\$0</code>	visualizza il nome del programma
<code>\$_</code>	visualizza lo stato dell'ultima chiamata del programma
<code>\$?</code>	visualizza lo stato di ritorno dell'ultima operazione di esecuzione di un processo esterno
<code>\$@</code>	visualizza l'ultimo messaggio di errore prodotto dallo script