

Perl

{ **re**começando com **perl**}

8.1 VARIÁVEIS ESCALARES

Agora que você já aprendeu a **fazer** algumas coisas muito úteis com a linguagem PERL, vamos começar realmente a estudá-la em detalhes, calmamente, em pequenas doses. Assim, você não só vai entender melhor o que aprendeu a fazer, como vai ter condições de escrever programas muito mais sofisticados que vão, por exemplo, criar campos em registros de arquivos, fazer leitura direta de um dado registro etc.

O primeiro conceito a aprender em PERL é o de **variável escalar**.

Na maioria das linguagens, quando se define uma variável é preciso, além de seu nome dizer que tipo de conteúdo (string, número etc.) ela vai ter. No PERL isso não é necessário. Você cria uma variável já definindo para ela um conteúdo inicial. E o tipo desse conteúdo vai definir o tipo da variável.

Assim, se criarmos:

```
$var1 = "TESTE";
```

a variável **var1** vai ser do tipo string.

Repare que todo nome de variável deve começar com o símbolo: **\$**.

8.2 DECIMAIS AMERICANOS

Nas operações numéricas a variáveis usarão sempre como separador de decimais (caso tenham decimais, claro) o símbolo **ponto** e não vírgula como é costume no Brasil.

Assim, podemos criar uma variável numérica assim:

```
$varn = 35.78;
```

Em arquivos devemos também sempre ter os valores de campos numéricos com decimais com esse separador. Logo, se a entrada de dados for feita com o separador **vírgula**, será preciso fazer uma substituição do símbolo (o que vamos aprender mais tarde).

Quando um valor numérico com decimais é projetado na tela, tem que ser feita a substituição inversa.

Só como curiosidade, fique sabendo que o PERL considera toda variável numérica como *float point* o que é uma decisão de arquitetura de linguagem discutível.

Não comece uma variável **numérica** com zero pois o PERL reserva isso para definir números na base *octal*. Assim, se definirmos:

```
$varo = 0377
```

teremos em **\$varo** o octal 0377 que equivale ao decimal 255. (Quem estudou ciências da

computação deve se lembrar dessas coisas. O prefixo 0X serve para hexadecimais: 0XFF, por exemplo é também 255).

8.3 ESCAPE DE LINHA SEGUINTE

Quando vamos imprimir (ou gravar) um string ou uma variável que é um string e queremos que depois dele impresso (ou gravado) se passe para a linha seguinte usamos o símbolo:
 \n

que deverá se colocado dentro das aspas assim:

```
"TESTE\n"
```

ou assim:

```
"$var1\n"
```

A variável será substituída pelo seu valor apesar de estar entre aspas, o que não aconteceria em outras linguagens.

É possível se circunscrever um string com apóstrofes no lugar de aspas. Isso, no entanto provoca que essas conversões não ocorram às vezes. Como toda linguagem, o PERL tem algumas atitudes estranhas, às vezes...