

# Perl

{operações básicas}

## 9.1 OPERAÇÕES NUMÉRICAS

Os operadores numéricos do PERL são os comuns na maioria das linguagens:

+	soma
-	subtração
*	multiplicação
/	divisão

Para a exponenciação temos: **\*\*** . Por exemplo:

```
4**2
```

seria 16.

Para achar o resto de uma divisão temos: **%**. Por exemplo:

```
9%2
```

seria 1.

Os operadores duplos, comuns no C++, são válidos no PERL. Use se você estiver acostumados com eles (++, +=, -- etc.)

## 9.2 ASSINALAMENTO E IGUALDADE

Você já deve estar cansado de saber isso mas vamos repetir. Para assinalamento no PERL temos o sinal **=**.

Isso não é equivalente a um sinal de igual da matemática. A expressão:

```
$x = $x + 2;
```

faria qualquer matemático ficar arrepiado (cortariamos \$x dos dois lados e teríamos que 2=0!).

Não confunda também assinalamento com o operador lógico que faz o teste de igualdade. No PERL temos DOIS operadores para este teste lógico. Tome cuidado com isso!

Para teste com numéricos o operador é:

```
==
```

Veja então:

```
$x = 5;
```

```
$y = 5;
```

```
$ret = ($x == $y);
```

No caso, em **\$ret** teremos **true**. Que é o retorno da operação lógica.

Para teste de igualdade entre strings o operador lógico é: **eq**. Assim teríamos:

```
$a = "ABC";  
  
$b = "DEF";  
  
$r = ($a eq $b);
```

O retorno, no caso seria **false**

Para testes de **desigualdade** temos também DOIS operadores:

```
!=  
ne
```

Sendo o primeiro para testes entre numéricos e o segundo para testes entre strings. Como já dissemos: não confunda!

Para comparações de tamanho entre numéricos temos os tradicionais:

```
<  
>  
<=  
>=
```

### 9.3 CONCATENAÇÃO DE STRINGS

A concatenação entre strings é feita com o operador **ponto**. Assim:

```
$s = "BOM"." "."DIA";
```

faria com que em **\$s** tivéssemos: BOM DIA (concatenados com um espaço).

As conversões entre numéricos e strings no PERL é feita automaticamente. Podemos escrever:

```
print "TOTAL = ".(4*5);
```

E teremos impresso o string: **TOTAL = 20**.

### 9.4 PRECEDÊNCIAS

A definição de precedências no PERL é feita com parênteses, como na maioria das linguagens.

Podemos colocar assinalamentos entre parênteses como em:

```
$b = 7 + ($a = 5);
```

Essa operação vai fazer com que a variável **\$a** fique com o valor 5 e a variável **\$b** fique com 12.

Evite, no entanto, essas linhas de códigos em que várias operações são feitas ao mesmo tempo. É verdade que isso faz com que alguns programadores se sintam um gênio. Às vezes isso realmente melhora a performance do programa. Mas o que acontece é que você não está escrevendo seus programas apenas para a máquina. Você também está escrevendo para o programador que vai ter que dar manutenção no seu programa amanhã. Assim sendo escreva de

maneira que ele entenda facilmente a seqüência das operações.

Não tente demonstrar que você é um gênio. Se você realmente for um não precisa mostrar. Minha avó, lá em Minas, sempre dizia: "Quem é realmente bom pode ser humilde..."