

## BASIC-256 :: Lição 3

### Variáveis e escolhas

Vamos introduzir agora alguns essenciais elementos de programação: variáveis, interação com o usuário e estruturas de escolha. Digite, execute e analise os programas a seguir.

#### programa 11: Variáveis numéricas

Variáveis são espaços de memória que podem ser acessados através de uma palavra. De forma mais simples, variáveis são nomes que damos a nossos dados para que possamos operar com eles através desse nome.

Vamos entender isso através de um exemplo. Suponha que você foi a um mercado e comprou uma dúzia de um certo produto que custa R\$ 20,00 a unidade. Você pode anotar isso em uma lista para calcular depois o quanto gastou:

```
quantidade = 12
preco = 20.00
```

Essa lista é, na verdade, exatamente a maneira que você escreverá em BASIC um programa que calculará o preço total. E para que o programa exiba o resultado, você deve escrever:

```
print "O total pago é " + quantidade*preco
```

Digite as três linhas e execute o programa. Tome cuidado com um erro frequente, que é se esquecer de fechar as aspas duplas, ou mesmo, caso você tenha copiado e colado o programa acima, ficar com a forma errada de aspas duplas em seu programa, originando uma mensagem de erro e impedindo sua execução.

Observe que calculamos o preço total multiplicando a quantidade pelo preço, usando os nomes que damos aos valores 12 e 20,00, como era de se esperar.

Observe também a presença do sinal de + indicando não uma soma, mas a concatenação da frase "O total pago é " com o resultado da operação da multiplicação.

Observe, por fim, que não usamos a cedilha no `c` da variável `preco`, e que o *basic256* utiliza um ponto no lugar da vírgula para marcar as casas decimais de um número.

Vejam agora como fazer quando desejamos dar nomes não a números, mas a outras palavras.

#### programa 12: Variáveis literais

Quando declaramos variáveis que contêm letras, o conteúdo deve estar entre aspas duplas, e um sinal de cifrão (\$) deve ser acrescentado no final do nome da variável:

```
filho$ = "Ricardo"
print "O nome do meu filho é " + filho$
```

Troque "Ricardo" por outro nome e reexecute o programa. Agora mude o programa para o seguinte:

```
filhos$ = "Ricardo, Henrique e Valquíria"
print "Os nomes dos meus filhos são " + filhos$
```

As variáveis literais servem também para guardar um conjunto de dados. Não só uma palavra, mas um texto inteiro.

### programa 13: Recebendo dados do usuário

Para interagir com o usuário, para atribuir a variáveis dados externos, use o comando `input`. Esse comando fará o computador esperar até que você digite algum valor na janela de texto:

```
input "Digite a quantidade: ", quantidade
preco = 20.00
print "O total pago é " + quantidade*preco
```

Se você deseja receber não um número, mas uma palavra ou mesmo uma frase inteira, use uma variável literal (variável com `$` no final):

```
input "Digite seu nome: ", nome$
print "Olá, " + nome$
```

### programa 14: Escolhendo entre alternativas

Escolher entre duas ou mais alternativas é uma tarefa comum em programação. O comando composto `if... then...` (se... então...) é usado quando desejamos comparar coisas ou decidir entre alternativas. Sua sintaxe é

```
if (condição) then (comandos)
```

Esse comando composto funciona da seguinte maneira:

1. se a condição logo depois do `if` for verdadeira, então ele executará os comandos depois do `then`
2. se a condição for falsa, ele passará para o restante do programa.

Vejamos um exemplo em que deixamos o computador decidir, a partir dos dados de entrada, se você é maior ou menor de idade.

```
input "Qual é a sua idade? ", idade
if idade >= 18 then print "Você é maior de idade."
if idade < 18 then print "Você é menor de idade."
```

A primeira condição foi `idade >= 18`, que lemos "idade maior ou igual a 18". Ou seja, se isso for verdadeiro, o programa imprimirá "Você é maior de idade." Se for falsa, ele passará para a comparação seguinte. Se esta for verdadeira, ele imprimirá "Você é menor de idade."

Os operadores que usamos para comprar coisas são: `>` (maior), `<` (menor), `>=` (maior ou igual), `<=` (menor ou igual), `=` (igual) e `<>` (diferente).

Se você deseja executar mais do que um simples comando depois do `then`, use `end if` para finalizar o bloco de comandos:

```

input "Qual é a sua idade? ", idade
if idade >= 18 then
    print "Você é maior de idade."
    print "Pode entrar"
end if
if idade < 18 then
    print "Você é menor de idade."
    print "Não pode entrar."
end if

```

Podemos melhorar esse programa. Observe que, em nosso caso, se uma pessoa não é maior de 18 anos, então, forçosamente, ela será menor de 18 anos. As alternativas são obviamente excludentes. Quando isso ocorre, podemos usar o comando `else` no lugar da segunda comparação. Se a primeira parte não for executada, a segunda o será, obrigatoriamente:

```

input "Qual é a sua idade? ", idade
if idade >= 18 then
    print "Você é maior de idade."
    print "Pode entrar"
else
    print "Você é menor de idade."
    print "Não pode entrar."
end if

```

### **programa 15: Jogando cara ou coroa**

O comando `rand` sorteia um número entre 0 e 1 (exceto o próprio 1). Fazendo uma variável receber o valor sorteado, podemos jogar cara ou cora no computador, estipulando, por exemplo, que temos cara quando o número sorteado for maior que 0,5, e coroa quando menor:

```

jogada = rand
if jogada >= 0.5 then
    print "Cara!"
else
    print "Coroa!"
end if

```

Cada vez que você aperta o botão *Run*, o computador sorteia um número. Aperte-o várias vezes e observe a caixa de texto. O computador está jogando cara ou coroa para você.