

Portugol

É uma linguagem de programação de alto nível (semelhante à linguagem natural, no caso, à língua portuguesa) desenvolvida por universidades do Brasil e de Portugal para fins didáticos.

Estrutura de um Programa

Um programa segue basicamente a estrutura:

```
programa
{

    funcao inicio(){

        // Aqui se escreve o código do seu programa. É comum colocar comentários
        nos códigos, para ativá-los, basta escrever duas barras

    }
}
```

Imprimir Frase

Para mandar o Portugol imprimir uma frase, utilizamos o comando **escreva()**, colocando dentro dos parêntesis, entre aspas, a frase que queremos que seja impressa. Por exemplo:

```
programa
{

    funcao inicio(){

        escreva("Olá mundo!")

    }
}
```

Ao compilar (rodar) o programa acima, sairá no console a frase: Olá mundo!.

Podemos fazer as quatro operações básicas utilizando os símbolos convencionais da aritmética: +, -, *, / (mais abaixo, há uma observação sobre a divisão).

Exemplo:

```
programa
{
    funcao inicio(){
        escreva("5 x 6 = ", 5*6)
    }
}
```

O que está acontecendo acima é uma concatenação, onde primeiro escrevemos o texto "5 x 6 = " e, em seguida, mandamos o Portugol imprimir o resultado da multiplicação 5*6. O resultado no console será "5 x 6 = 30".

Variáveis

Para declarar uma variável, devemos especificar o tipo dela: **inteiro**, se ela assumir valores inteiros; **real**, se ela assumir valores reais; **caracter**, se o valor dela for um único símbolo; **cadeia**, se o valor for um conjunto de símbolos, ou seja, uma palavra ou texto.

Em programação, é comum darmos um nomezinho para uma variável, diferentemente da matemática, onde as variáveis são representadas simplesmente por uma letra.

Exemplo:

```
programa
{
    funcao inicio(){
        inteiro numero
    }
}
```

Acima, declaramos uma variável para armazenar um número inteiro, e a batizamos de `numero`.

Se quisermos que o usuário atribua um valor à variável, utilizamos o comando **leia()**.

Exemplo de código que recebe um número inteiro digitado pelo usuário e devolve o sucessor desse número:

```

programa
{

    funcao inicio(){

        inteiro numero
        escreva("Digite um número inteiro: ")
        leia(numero)
        escreva("O sucessor do número digitado é: ", numero+1)

    }

}

```

Operadores e Desvios Condicionais

Para compararmos dois valores numéricos, utilizamos os operadores relacionais ==, !=, <, <=, >, >= (igual, diferente, menor, menor ou igual, maior, maior ou igual, respectivamente). Observação: no Portugol, utiliza-se dois sinais de igual seguidos para simbolizar a relação "é igual a", pois um sinal de igual apenas é utilizado para uma variável receber um valor.

Temos também os operadores aritméticos: div (divisão inteira, por exemplo, $10/3 = 3$ no Portugol) e mod (resto da divisão, por exemplo, $10\%3 = 1$).

Por fim, temos os operadores lógicos (também chamados de conectivos lógicos, por conectarem duas ou mais condições criadas com os outros operadores): ou, e.

Lembrando que: p ou q é verdadeiro quando pelo menos uma das condições é verdadeira; enquanto p e q é verdadeiro somente quando ambas as condições são verdadeiras.

Podemos utilizar o desvio condicional **se()** para verificar se uma condição está sendo satisfeita ou não.

```

programa
{

    funcao inicio(){

        inteiro numero
        escreva("Digite um número inteiro: ")
        leia(numero)
        se(numero%2 == 0){
            escreva("O número ", numero, " é par")
        }senao{
            escreva("O número ", numero, " é ímpar")
        }

    }

}

```

```

    }
}

```

Laços de Repetição

Para abreviar coisas repetitivas, podemos utilizar os laços de repetição: **para()** e **enquanto()**.

```

programa
{
    funcao inicio()
    {
        inteiro n,i
        escreva("Digite um número: ")
        leia(n)
        para(i=1; i<=n; i++){
            escreva(i, "\n")
        }
    }
}

```

O código a seguir faz a mesma coisa que o anterior:

```

programa
{
    funcao inicio()
    {
        inteiro n, i=1
        escreva("Digite um número: ")
        leia(n)
        enquanto(i<=n){
            escreva(i, "\n")
            i = i+1
        }
    }
}

```