

# Mestrado Decisão Económica e Empresarial COMPUTAÇÃO

Programação em VBA.

Operadores e operandos. Estruturas de controlo.

Exercícios.

# Variáveis/ Constantes

---

Tal como a variável

- tem um nome alfanumérico
- deve ter um tipo de dados

**Const Nome As Tipo = expressão**

- É usado como um sinónimo, se houver necessidade de alteração é mais fácil

Const IVA As Single = 0.23

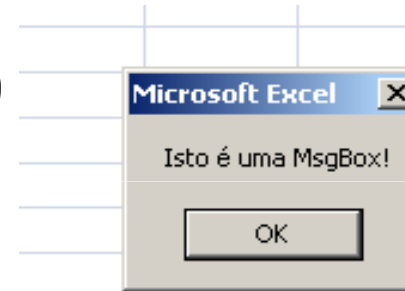
# Objectos gráficos

A forma mais simples do Macro interagir com o utilizador é através de duas funções:

- MsgBox

`MsgBox("Isto é uma MsgBox!")`

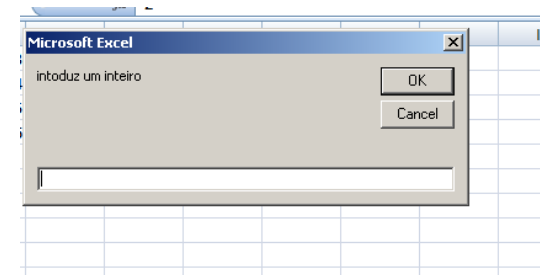
`MsgBox (Mensagem, características, Título)`



- InputBox

`A = InputBox("intoduz a idade")`

`InputBox (Mensagem,Título, valor_de_defeito,xpos,ypos)`



# Operadores e operandos

---

- Uma operação é uma acção executada com um ou mais valores para modificar um valor ou para obter um novo valor por combinação de valores existentes.

$$y+z$$

- Numa operação existe pelo menos um símbolo e um valor.
- O símbolo é o **operador +**
- O valor ou variável é um **operando y,z**

# Operadores e Operandos

---

---

- Os operadores podem ser
  - unários  $-X$  ( envolvem apenas um operando)
  - Binários  $X+Y$  ( envolvem dois operandos)

Dim (para declarar variáveis) é também um operador

# Operadores

---

- Afecção =
- Continuação de linha \_
- Parêntesis ( )
- Vírgula , `Dim A As Integer, B As Integer`
- Aspas “ “ ”texto” string
- Dois pontos : `Sub Exemplo() : End Sub`
- Concatenação de strings `& "texto1" & "texto2"`
- Mudança de linha `vbCrLf`

# Operadores Aritméticos

---

- Unários + e –
- Adição +
- Subtracção –
- Multiplicação \*
- Potenciação ^
- Divisão decimal /
- Divisão Inteira \
- Resto da divisão inteira Mod

# Operadores lógicos

---

- Menor <
- Maior >
- Menor ou igual <=
- Maior ou igual >=
- Igualdade =
- Desigualdade <>

- Conjunção **And**
- Disjunção **Or**
- Disjunção exclusiva **Xor**
- Negação **Not**



# Estruturas de controlo - If...Then...Else

---

**If** condição **Then**

instruções

**Else**

instruções

**End If**

# Estruturas de controlo - If...Then

---

**If** condição **Then**

instruções

**End If**

# Estruturas de controlo (encadeadas1)

```
If condição1 Then  
    instruções1  
Elseif condição2 Then  
    instruções2  
Elseif condição3 Then  
Else  
    ...  
    instruções n  
End If
```

```
If (nota<0) Or (nota>20) Then  
    Resultado="Nota Inválida!"  
ElseIf (nota<6) Then  
    Resultado="Mau"  
ElseIf (nota<10) Then  
    Resultado="Mediocre"  
ElseIf (nota<14) Then  
    Resultado="Suficiente"  
ElseIf (nota<17) Then  
    Resultado="Bom"  
Else  
    Resultado="Muito Bom"  
End If
```

# Estruturas de controlo (encadeadas2)

```
If condição1 Then  
    instruções1  
Else  
    If condição2 Then  
        instruções2  
    Else  
        If condição3 Then  
            instruções3  
        Else  
            instruções4  
        End If  
    End If  
End If
```

# Estruturas de controle - selection ...case

**Select case** expressão

**Case** instância  
instrução1

**Case** condição  
instrução2

**Case** Range Is  
instrução3

**Case** Else  
instrução4

**End Select**

```
Select Case nota
  Case 0 To 6
    resultado = "Mau"
  Case 7 To 9
    resultado = "Mediocre"
  Case 10 To 13
    resultado = "Suficiente"
  Case 14 To 16
    resultado = "Bom"
  Case 17 To 20
    resultado = "Muito Bom"
End Select
```

```
Select Case nota
  Case Is > 10
    resultado = "Aprovado"
  Case Is < 10
    resultado = "Reprovado"
  Case Else
    resultado = "Tangente!"
End Select
```

# Estruturas de Controlo

**Do While** condição  
instruções

**Loop**

```
Contador=1
Soma=0
Do While contador <= n
    Soma=Soma+(contador^2)/3
    contador=contador+1
Loop
```

**Do**  
instruções

**Loop While** condição

```
Contador=n
Fact=1
Do
    Fact=Fact*contador
    contador=contador-1
Loop While contador > 1
```

# Estruturas de Controlo

**Do Until** condição  
instruções

**Loop**

```
contador = 1
Soma = 0
Do Until contador > n
  Soma = Soma + (contador ^ 2) / 3
  contador = contador + 1
Loop
```

**Do**  
instruções

**Loop Until** condição

```
Contador=0
Fact=1
Do
  contador = contador + 1
  Fact = Fact * contador
Loop Until contador >= n
```

# Estruturas de Controlo

**For** contador= v\_inicial **To** v\_final  
instruções  
**Next** contador

```
For contador = 1 To n  
    Soma = Soma + contador ^ 2 / 3  
Next contador  
For contador = 1 To n Step 2  
    Soma = Soma + contador ^ 2 / 3  
Next contador
```

**For** contador= v\_inicial **To** v\_final **Step** passo  
instruções  
**Next**

**For Each** objecto numa colecção  
instruções  
**Next**



# Estruturas de Controlo (encadeadas)

---

**For** contador1= v\_inicial1 **To** v\_final1

**For** contador2= v\_inicial2 **To** v\_final2

instruções

**Next** contador2

**Next** contador1

```
Dim i As Integer
Dim j As Integer
For i = 1 To n
  For j = 1 To n
    Worksheets("Sheet1").Cells(i, j).Value = i + j
  Next j
Next i
```

# exercícios

1. Escreva um macro que escreve 3 inteiros por ordem decrescente.
2. Programe um macro para determinar o custo de uma encomenda sabendo que o preço base é de 100 e que é efectuado um desconto em função da quantidade de acordo com a seguinte tabela:

quantidade	desconto
$\geq 150$	10%
$\geq 300$	15%
$\geq 500$	20%
$\geq 1000$	25%

3. Escreva um macro para determinar se um número é primo.
4. Escreva um macro para determinar o máximo divisor comum.
5. Escreva um macro que informa se um número é par ou ímpar