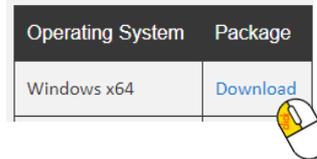




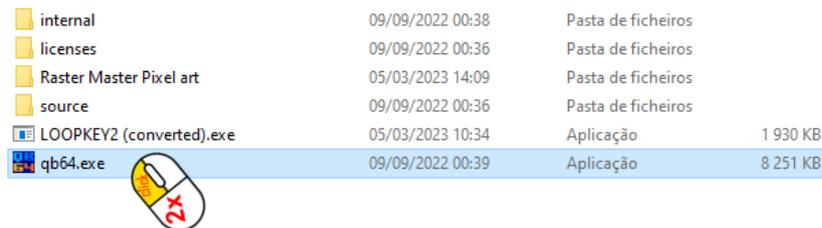
 Neste guião irás aprender a criar **subrotinas** e **funções** em **QBASIC64**.

 Começa por descarregar o **QB64** [aqui](#)



Escolhe a **última versão** disponível e **instala-o** no teu **pc** num local à escolha.

 **Abre o QB64.**

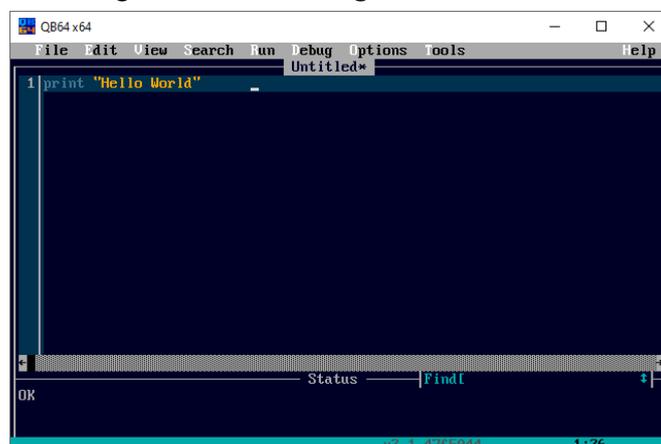


 A **interface do programa** será **mostrada** numa **nova janela** no **ecrã**:



 A **interface** é muito **semelhante** à do **QBASIC 4.5**, mas possui várias **vantagens** como se explicou na aula teórica.

 Começa por **adicionar** a seguinte **linha de código**:





 Como podes **verificar** o **código** fica **pintado** de cor **distinta** da **versão anterior** que permite uma melhor **leitura/interpretação** do código.

 Pressiona **F5** para veres o **resultado** deste **programa**. O **programa** dá um **aviso** que irá criar um **executável** e o **guardará** na **pasta** onde está o **QB64**. **Prossegue**.



Se **reparares** na pasta do **QB64** apareceu lá um **ficheiro** novo de nome **untitled.exe**. Isto porque **não atribuímos** nenhum **nome** ao **programa**. Vamos guardar como **1QB64.bas** e **voltar** a **correr** o programa.

Se **reparares** na pasta ficas com o **dois** novos **ficheiros** o **1QB64.bas** e o **1QB64.exe**

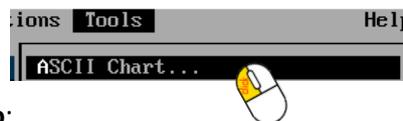
1QB64.bas	11/03/2023 09:29	Ficheiro BAS	1 KB
1QB64.exe	11/03/2023 09:29	Aplicação	1 930 KB

 Em qualquer altura podes **correr** o teu programa **diretamente** através de **duplo click** sobre o ficheiro **exe** não havendo **necessidade** de entrar no **QB64** e **compilar** o **programa** (**poderás ter de criar uma exceção no Windows Defender**).

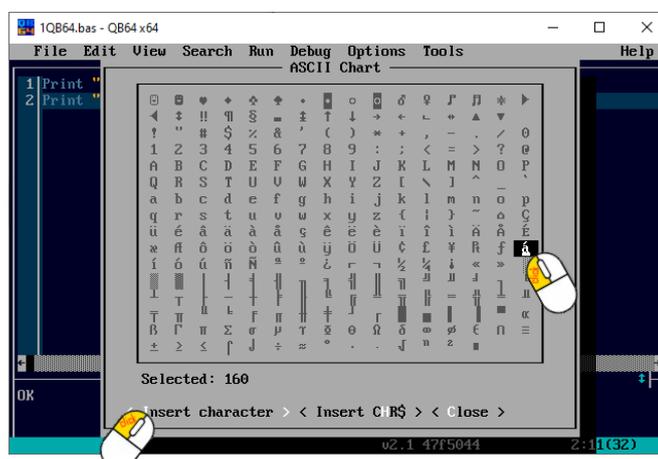
 Vamos agora **adicionar** uma **nova linha** ao **programa**

Print “Olá mundo”

Não consegues inserir a com acento, correto? Este é um dos **problemas** deste **IDE**, mas que **facilmente** se **ultrapassa**. Acede ao menu **Tools**→**ASCII Chart**



 E **adiciona** o **á** com **acento**:





O resultado será:

```
1 Print "Hello World"
2 Print "Olá_mundo"
```

Se **correres** o **programa** irás ver o á com acento. Verifica:

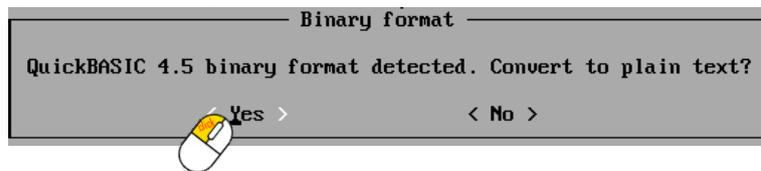
```
Hello World
Olá mundo
```

Vamos então começar a **trabalhar** com **subrotinas** e **funções**.

Abre o ficheiro do último guião **inloop.bas** (se não souberes abrir faz uma cópia pelo explorador de ficheiros do Windows do programa para a pasta QB64):

```
1 - Jogar
2 - Sair
```

Ao **abrir** pede para **converter** para **texto**. Diz que **sim**.



Ficarás com algo assim:

```
1 Cls
2 Screen 13
3 Color 7
4 Print "*****"
5 Print "1 - Jogar"
6 Print "Q - Sair"
7 Print "*****"
8 Do
9   If InKey$ = "1" Then
10    Cls
11    Print "Bem vindo ao Jogo"
12   End If
13 Loop Until InKey$ = "q"
14 System
```

Corre o programa.

Funciona? Mas tem **Lag!** Nem **sempre** a **tecla** parece ser **lida**, não é?

Podemos **arranjar** este **problema**. Começa pode **definir** uma **variável**:

```
DIM KeyPress AS STRING
```

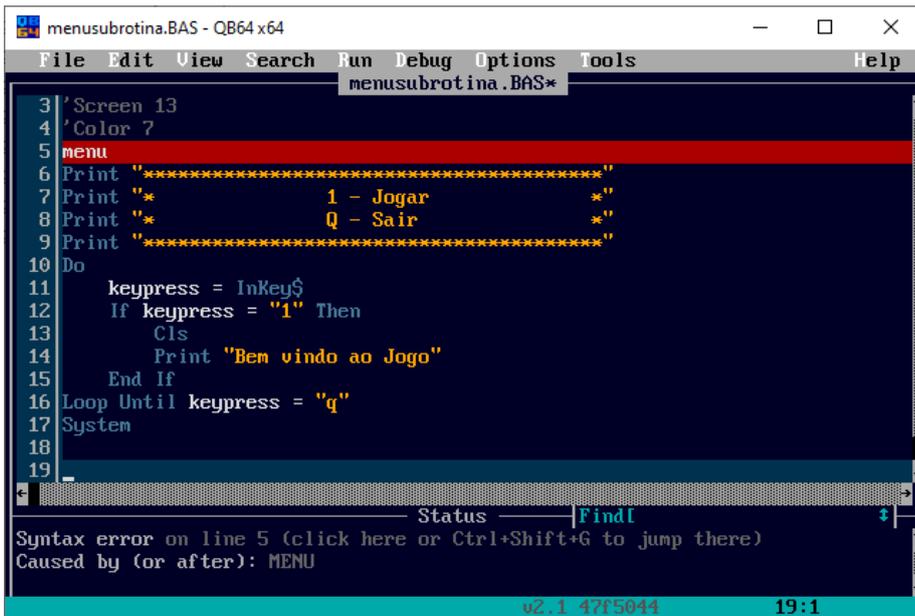


 Agora atribui a **KeyPress** o valor de **INKEY\$**. **Substitui** todos os locais onde utiliza **INKEY\$** pela variável **KeyPress**, inclusive na **condição de paragem** e verás que já irá funcionar **sem Lag**.

 **Retira** o screen 13 para a janela não ficar tão pequena.

 **Adiciona** um **subrotina** para que seja ela a **imprimir** o menu.

 **Começa** por fazer o seguinte. No local onde está o menu **digita** apenas a **palavra menu**.

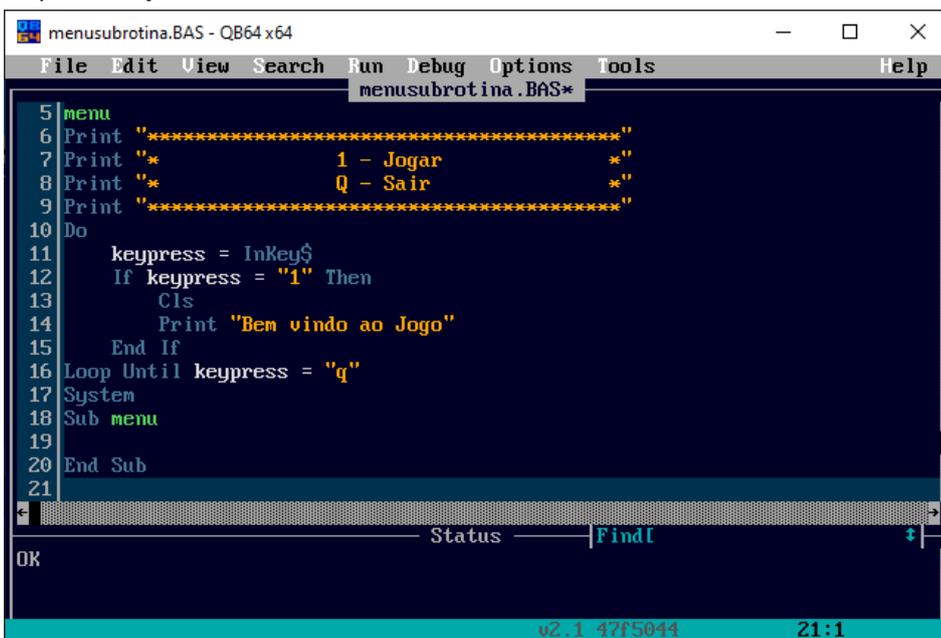


```
3 'Screen 13
4 'Color 7
5 menu
6 Print "*****"
7 Print "*           1 - Jogar           *"
8 Print "*           Q - Sair           *"
9 Print "*****"
10 Do
11   keypress = InKey$
12   If keypress = "1" Then
13     Cls
14     Print "Bem vindo ao Jogo"
15   End If
16 Loop Until keypress = "q"
17 System
18
19
```

Syntax error on line 5 (click here or Ctrl+Shift+G to jump there)
Caused by (or after): MENU

v2.1 47f5044 19:1

 Ficarás com um **erro**. Isto porque o **compilador** ainda **não sabe** o que é **menu**. Na **última linha**, depois de **system** vamos criar a **subrotina menu**.



```
5 menu
6 Print "*****"
7 Print "*           1 - Jogar           *"
8 Print "*           Q - Sair           *"
9 Print "*****"
10 Do
11   keypress = InKey$
12   If keypress = "1" Then
13     Cls
14     Print "Bem vindo ao Jogo"
15   End If
16 Loop Until keypress = "q"
17 System
18 Sub menu
19
20 End Sub
21
```

OK

v2.1 47f5044 21:1

 **Falta conteúdo** na sub. O que será? Pois, **todo** o menu. **Corta** e **cola** o menu (linha 6 a 9) e **cola** dentro da sub.



 Ficarás com algo do género:

```
menusubrotina.BAS - QB64 x64
File Edit View Search Run Debug Options Tools Help
menusubrotina.BAS*:menu
5 menu
6 Do
7   keypress = Inkey$
8   If keypress = "1" Then
9     Cls
10    Print "Ben vindo ao Jogo"
11  End If
12 Loop Until keypress = "q"
13 System
14 Sub menu
15   Print "*****"
16   Print "*           1 - Jogar           *"
17   Print "*           0 - Sair            *"
18   Print "*****"
19 End Sub
20
21
Status Find
v2.1 4775044 18:53
```

 Se **correres** o **programa** o **resultado** será o **mesmo**, mas com um **código** mais **organizado** e com a **vantagem** de poderes **chamar** de **novo** o **menu** apenas com a palavra **menu**.

 Guarda o **programa** como **menusubrotina.bas** (este IDE já **aceita** mais de **8** **carateres**).

 Abre o programa **oddrnd.bas**

 Corrige o **português** (**acentos** e **carateres** **especiais**).

```
"Nº nfo divisível por 2. Nº ímpar."
```

 Pretende-se que seja **possível** através de uma **subrotina** dizer se o **nº** **introduzido** pelo **utilizador** é **par** ou **ímpar**. Começa por **criar** uma **subrotina** de nome **odd** que **aceite** **1** **parâmetro**. Esse **parâmetro** deve ser o que o **valor** que **utilizador** **introduziu** e vai ser **testado** dentro da **subrotina** mostrando no **ecrã** **par** ou **ímpar** conforme o **caso**.

 Guarda o teu **programa** como **oddsb.bas**

 Cria um **novo** **programa**.

 Neste **programa** iremos **utilizar** uma **função** para **calcular** as **operações** **básicas** (+, -, /, *) **entre** dois **números** **introduzidos** pelo **utilizador**. Começa por **digitar** o seguinte **código**.

```
1 Cls
2 Dim num1 As Single
3 Dim num2 As Single
4 Dim operacao As String
5 Input "Introduza um número: ", num1
6 Input "Introduza outro número: ", num2
7 Input "Introduza a operação pretendida (+, -, /, *): ", operacao
8 Print "O resultado é"; op(num1, num2, operacao)
```

Neste código são **definidas** **duas** **variáveis** onde são **guardados** dois **números** **introduzidos** pelo **utilizador**. De seguida é chamada uma **função** de nome **op** que recebe **3** **parâmetros**.

 Cria a **função** **op** onde conforme o **terceiro** **parâmetro** da **função** faz a **soma**, **subtração**, **divisão** ou **multiplicação** dos **números** inseridos nos **parâmetros** **1** e **2**. Relembro a **sintaxe** de uma **função**:



FUNCTION **nome** (parametron_1 as tipo, parametron_2 as tipo, ...,parametron_n as tipo)

<aqui as instruções>

nome = o valor que queremos retornar

END FUNCTION

 **PRO:** “Em vez do resultado é” cria uma **subrotina** que imprima a frase:

O resultado da “operacao” é:

No **lugar** de **operacao** deve ser escrito **soma**, **subtração**, **divisão** ou **multiplicação** conforme o **sinal** da operação **inserido** pelo **utilizador**.

 **Guarda** o teu programa como **funcaooperadores.bas**

 **Chama** o teu **professor** para **avaliar**.