



#### Unidade 2: Iniciação à programação em Python

#### Lição 1: Funções condicionais

Nesta primeira lição da Unidade 2 vamos descobrir como escrever e utilizar uma função condicional em Python.

#### Objetivos:

- Escrever e utilizar uma função condicional
- Revisitar o conceito de função em Python

Num programa é muito frequente ter que se estruturar a sua execução em função de **condições** que envolvem diferentes variáveis. Uma condição é uma expressão que pode tomar o valor lógico **verdadeiro** ou **falso**, conforme o(s) valor(es) atribuído(s) à(s) variável(eis).

Por exemplo, “**a = b**” ou então “**a ≥ b**”, ou ainda “**n é par**”, são exemplos de condições que são verificadas pela concretização das variáveis por alguns valores: as soluções.

Assim, num programa podemos testar uma condição verificando se ela se transforma numa proposição verdadeira ou numa proposição falsa, conforme concretizamos a variável com um certo valor ou um outro qualquer. Designamos este processo por **verificação condicional**.

```

if condição :
    Instrução A
else :
    Instrução B

```

Estrutura de uma função condicional

#### OBSERVAÇÕES:

- Na linguagem Python não existe nenhuma instrução para indicar o fim da função condicional. É, mais uma vez, a indentação que desloca as instruções condicionais A e B para a direita, definindo assim os blocos de instrução da função condicional.
- Utiliza-se **elif** como contração de **else if** .
- Para testar a igualdade entre dois valores, na linguagem Python, utiliza-se a sintaxe “**=**” (dois sinais de igual seguidos).

#### EXEMPLO:

Uma empresa de aluguer de veículos apresenta aos seus clientes o seguinte contrato:

Um custo base de 66€ ao que se adiciona o custo de 0,25€ por cada quilómetro percorrido a partir dos 70 quilómetros.

Cria um programa que permita calcular automaticamente o custo, C, do aluguer de um veículo para uma dada viagem em função da distância percorrida, X quilómetros.

```

ALGORITMO
Entrada: a (distância percorrida)
Saída: C – custo aluguer
Procedimentos:
X ← a
Se 0 < X < 70 então
    C ← 66
Senão
    C ← 66 + 0.25*(X-70)
FimSe
Escrever C

```

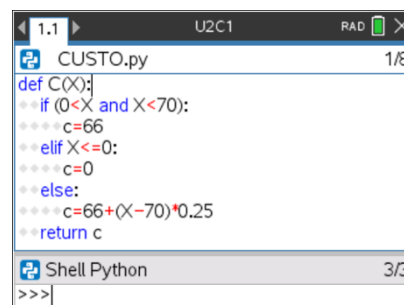
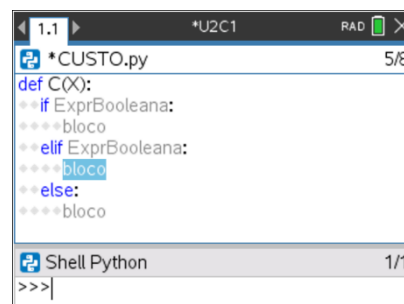
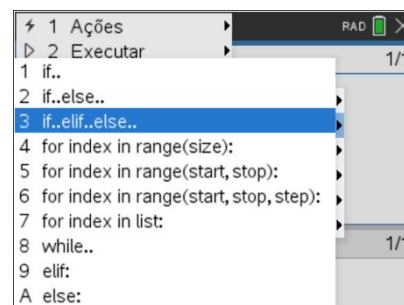
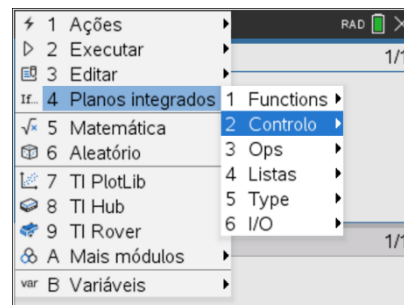
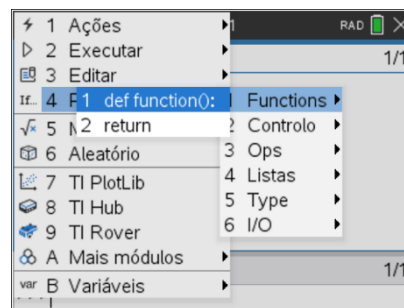
#### OBSERVAÇÃO:

Acautele o eventual caso de o utilizador atribuir um número negativo à variável X, isto é, garanta o domínio da variável X.



### IMPLEMENTAÇÃO DO ALGORITMO:

- Abra um novo documento da TI-Nspire CX II T.
- Crie uma nova página com o editor de programas da aplicação Python, designe o programa por “CUSTO” e de seguida pressione **enter** para validar.
- Divida a página em duas aplicações (editor Python, interpretador Python), esquema de página na horizontal, como já realizado na Unidade 1.
- Na página do editor de Python, clique na tecla **menu**, depois selecione a opção **4: Planos integrados**, seguida da opção **1: Funções** e, por fim, a opção **1: def function()**.
- Designe a função por C(X), função de um argumento.
- De seguida clique na tecla **menu**, depois selecione a opção **4: Planos integrados**, e por fim a opção **2: Controlo**.
- Selecione, por fim, a função condicional **3: if ... elif ... else**.
- Observe a indentação automática da estrutura que constitui a função condicional selecionada.
- Complete a função condicional baseando-se no algoritmo proposto para esta tarefa.
- Utilize a tecla **tab** para aceder mais facilmente a cada campo, **ExprBooleana** ou **bloco**, da função condicional.
- Deverá obter o programa que se encontra na figura ao lado.
- Os símbolos operacionais **<** e **≤**, e também o operador lógico **and**, podem obter-se através da opção **3:Ops** do submenu **4:Planos integrados**. Poderão, também, ser facilmente obtidos através do teclado da unidade portátil, pressionando sucessivamente as teclas **ctrl** e **=**.
- Pressione simultaneamente as teclas **ctrl** e **B** para verificar a sintaxe e guardar o programa.





- Execute o programa, clicando simultaneamente nas teclas **ctrl** e **R** ou colocando o cursor no interpretador (Shell Python), para determinar o custo de uma viagem com um veículo alugado nesta empresa.

```

1.1 *U2C1 RAD
CUSTO.py 8/8
c=0
else:
c=66+(x-70)*0.25
return c

Shell Python 4/5
>>>#Running CUSTO.py
>>>from CUSTO import *
>>>C(10)
66

```

#### OBSERVAÇÃO:

O atalho **ctrl** e **B** guarda a atualização do programa construído na aplicação TI-Python, mas não guarda o documento tns. Para guardar o documento tns, com todas as páginas que o compõem, deve usar o atalho teclas **ctrl** e **S** ou usar a opção adequada após pressionar a tecla **doc**.

#### APLICAÇÃO DAS APRENDIZAGENS:

##### Funções definidas por ramos

Seja  $f$  uma função, real de variável real, definida por ramos por:

$$f(x) = \begin{cases} 2x + 1 & \text{se } x \leq -1 \\ -x + 2 & \text{se } -1 < x \leq 0 \\ -3x + 2 & \text{se } x > 0 \end{cases}$$

Crie um programa, conforme imagem ao lado, e use-o para poder completar a tabela abaixo:

x	-4	-1.5	-0.5	-0.1	0.6	2.5	4.8	7.3
f(x)								

Execute o programa.

```

1.1 1.2 *U2C1 RAD
func_ramos.py guardado com sucesso
def f(x):
if x <= -1:
f=2*x+1
elif x <= 0:
f=-x+2
else:
f=-3*x+2
return f

Shell Python 1/1
>>>

```

```

1.1 1.2 *U2C1 RAD
func_ramos.py 3/8
def f(x):
if x <= -1:
f=2*x+1
elif x <= 0:

Shell Python 6/7
>>>#Running func_ramos.py
>>>from func_ramos import *
>>>f(-0.5)
2.5

```

