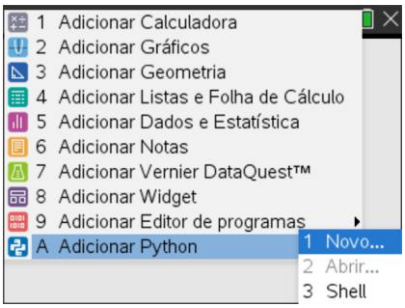
```
texto=input("Usando só letras minúsculas, introduza a frase: ")
def rot13(texto):
    l=len(texto)
    code=""
    for i in range(l):
        c=ord(texto[i])
        c=c+13
        if c>122:
           c=c-26
        m=chr(c)
        code=code+m
    return code
print(rot13(texto))
```

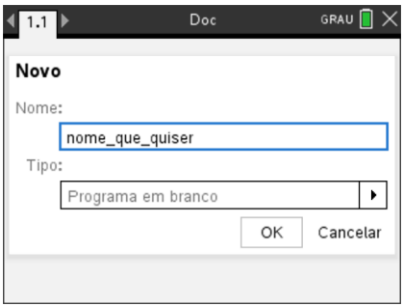
Editar Python na TI-nspire CX II-T

Ligue a sua calculadora e crie um novo documento.
Escolha uma página de *Python*:

menu A **Adicionar Python** → 1 **Novo.**

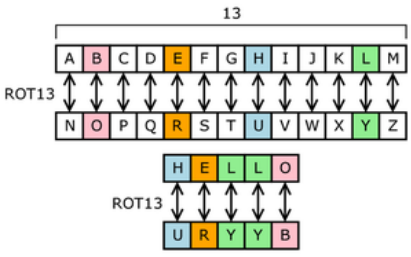
Coloque um nome à sua escolha, de seguida, prime em **OK**.
Abre-se uma página vazia, que é o editor de *Python* da calculadora/tecnologia TI-Nspire CX II-T, onde deve escrever o código.





1. Como encriptar um texto, recorrendo à sua conversão em código ASCII, seguido da adição de um valor de rotação?

O programa seguinte tem como objetivo encriptar mensagens utilizando uma cifra do tipo César – conhecida como ROT13 (*Rotation 13*). Trata-se de um algoritmo de substituição simples que opera sobre o alfabeto latino, deslocando cada letra 13 posições à frente no alfabeto. Como o alfabeto possui 26 letras, aplicar ROT13 duas vezes devolve a mensagem original, tornando-o um método simétrico.



Fonte: https://pt.wikipedia.org/wiki/ROT13

Por exemplo, se o utilizador colocar "ola" para encriptar, segundo a Cifra Rot-13:

o → b **l → y** **a → n** ⇉ **ola → byn**

No programa, o utilizador introduz uma frase composta apenas por letras minúsculas, e o algoritmo aplica ROT13 a cada caráter individualmente. Para isso, utiliza-se a tabela ASCII (*American Standard Code for Information Interchange*) — uma codificação padrão que atribui um número a cada caráter, como letras, números, pontuação e símbolos. Neste caso serão utilizadas apenas letras minúsculas:

Código ASCII																								
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	121

I. Pretende-se fazer um programa que, depois de executado, comece por pedir ao utilizador o texto, apenas constituído por letras minúsculas, para encriptar.

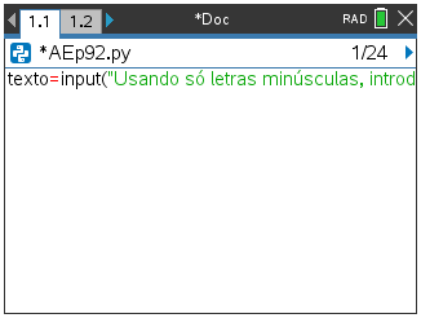
Em *Python*, a função que permite receber um dado por parte do utilizador é **input()**.

O pedido para o utilizador introduzir a frase é feito através de uma mensagem:

input("Usando só letras minúsculas, introduza a frase:")

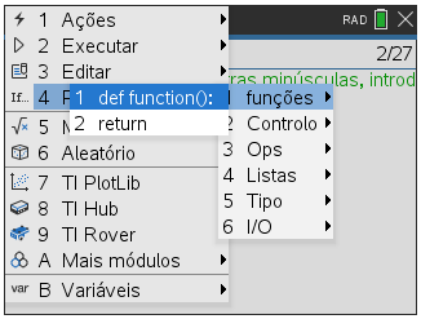
Após o utilizador introduzir a frase, esta será guardada na variável **texto**.

texto= input("Usando só letras minúsculas, introduza a frase:")



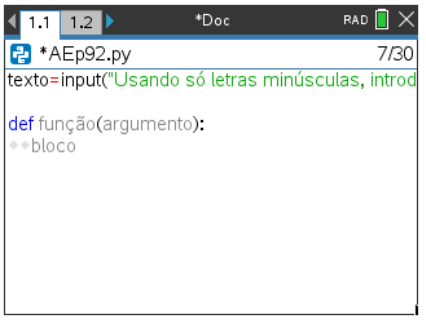
II. Com o objetivo de se encriptar texto usando a cifra ROT13, convém definir a função rot13, que irá aplicar a cifra à ao texto introduzido pelo utilizador e, no fim, devolve o resultado cifrado.

Esta "função" é semelhante ao que noutras linguagens de programação se chama de subrotina, ou seja, uma secção de código que é ativada a partir das linhas de código do programa subsequentes. Esta estrutura é bastante útil, por exemplo, quando no programa se precisa de fazer algumas vezes as ações que nela estão previstas. A ideia desta estrutura é oportuna num contexto de funções matemáticas e, por isso, surge aqui como exemplo de um passo mais à frente em relação à simples utilização de uma estrutura de condição.



Para obter esta estrutura, caso conheça a sintaxe, poderá escrever com o teclado, mas pode também obter através do menu:

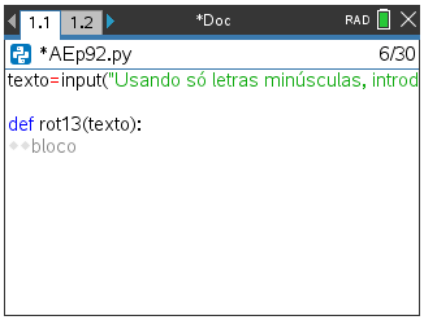
menu 4 **Planos integrados** → 1 **funções** → 1 **def function():**



Obtida a estrutura, é agora necessário preencher os campos em falta.

A função, chamada de **rot13**, é uma função que recebe uma *string* (uma cadeia de caracteres) (**texto**) como argumento:

```
def rot13(texto):
    bloco
```



Este código pode ser facilmente escrito com o teclado, ou obtida no menu, na mesma lista pendente onde se obteve a função **def function():**. Da estrutura base, deve apagar-se o que não for necessário para esta secção funcionar conforme o pretendido.

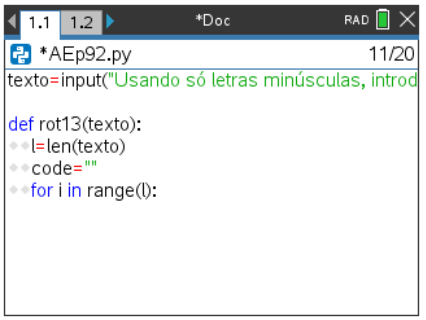
III. No programa, a cifra ROT13 irá encriptar, um a um, todos os caracteres do texto introduzido. Para tal, será necessário utilizar um ciclo de repetição **for** que percorra todos as posições da frase, de 0 até ao comprimento da frase menos 1.

Em *Python*, para saber o comprimento da frase utiliza-se a função **len()**.

Esta pode ser escrita com o teclado, mas também pode ser obtida através do menu:

menu → 4 **Planos integrados** → 4 **Listas** → 3 **len()**

Para além disto, à medida que os caracteres forem sendo encriptados, estes vão sendo guardados numa variável, designada por **code**, que inicializa vazia, sem espaços.



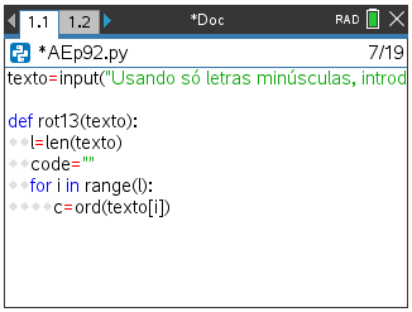

```
def rot13(texto):
    l=len(texto)            # o comprimento da frase é l
    code=""                 # a variável começa vazia, sem espaços
    for i in range(l):      # para cada caráter da frase da posição 0 até à
                              à posição l-1
```

IV. Como já foi referido, o programa vai encriptar cada letra do texto introduzido pelo utilizador. Assim, na letra da frase que está na posição i, **texto[i]**, o programa vai convertê-la para o seu valor ASCII (por exemplo, como se pode visualizar na tabela "Código ASCII", a letra a é convertida no valor 97.)

Em Python, esta conversão é feita usando a função **ord()**. Esta função não se encontra no menu, tendo que ser escrita com o teclado.

Assim, deve-se inserir, dentro do ciclo **for**, inserir a seguinte linha de código:

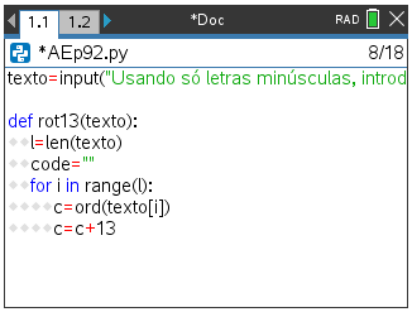
c=ord(texto[i])



V. Após a letra ter sido convertida para ao seu valor ASCII, é adicionado 13 a este valor para efetuar o deslocamento respetivo à cifra ROT13.

Portanto, acrescenta-se a linha de código seguinte, dentro do ciclo **for**:

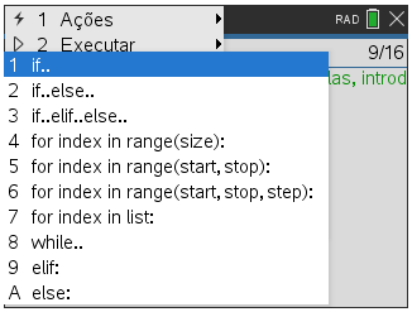
c=c+13



VI. **Se**, ao efetuar esta adição, o valor ultrapassar o 122, que é o código ASCII da letra "z", subtrai-se 26 para que a contagem volte ao início do alfabeto, respeitando o comportamento circular deste.

Para tal, em *Python*, é necessário utilizar uma estrutura condicional, **if**, que pode escrever com o teclado ou obter no menu:

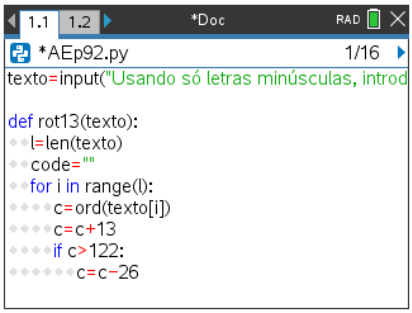
B → 4 **Planos integrados** → 2 **Controlo** → 1 **if..**



Assim, tem-se de acrescentar as seguintes linhas de código (ainda dentro do ciclo **for**):

if c>122:

c=c−26



VII. Feito o deslocamento, o programa precisa de converter o novo valor numérico de volta para uma letra. Esta conversão é feita através da função **chr()**. Esta é outra função que não se encontra no menu, tendo que ser escrita com o teclado.

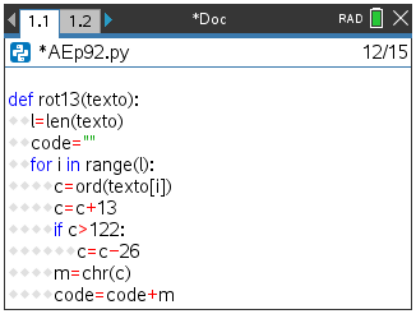
Assim, ainda dentro do ciclo **for**:

m=chr(c)

Esta letra é a nova letra encriptada que será adicionada à mensagem cifrada, **code**.

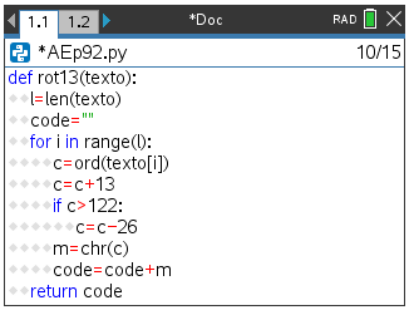
Assim, a linha de código que se segue é a seguir:

code=code+m



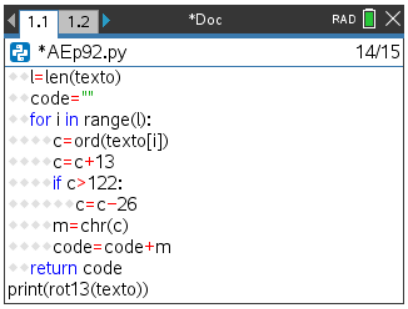
VIII. Depois de terminar a execução do ciclo de repetição, ou seja, quando for percorrido todo o texto introduzido pelo utilizador, espera-se que a função retorne a frase codificada, **code**. Portanto, já fora do ciclo, deve-se escrever **return** com o que se pretende que seja o resultado, a saída desta secção de código:

return code

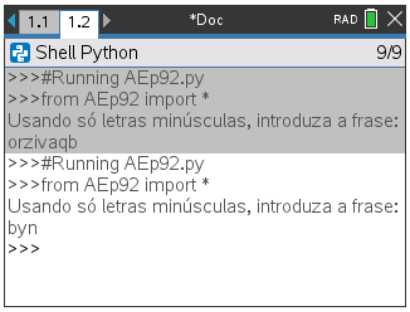


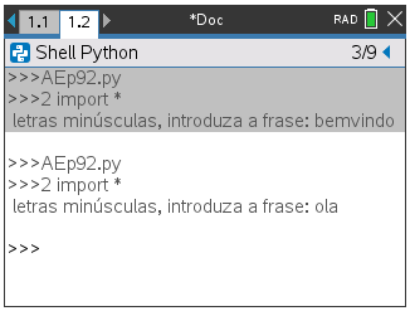
IX. De maneira a concluir o programa é necessário colocar as instruções que permitem a observação ou a impressão da informação pretendida, ou seja, da do texto "**texto**", encriptado pela cifra ROT13, **rot13(texto)**, na página Shell Python, que se abre automaticamente logo após a execução do programa.
Deste modo, insere-se a seguinte linha de código:

print(rot13(texto))



X. Escrito o programa, falta executá-lo. Pode utilizar-se uma instrução do menu (menu 2 1), mas é claramente mais simples utilizar um atalho, uma combinação de teclas (ctrl + R).
Neste caso, como se pode observar nas imagens ao lado, é preciso ter atenção que, ao haver uma frase introdutória muito grande, quando se coloca a frase que se pretende encriptar, ela irá "sair" fora do display.
O resultado aparece numa nova página destinada a mostrar o resultado da execução do programa, ***Shell Python***, na qual também se podem fazer operações e pequenos programas, mas que não permanecerão gravados após o fecho da aplicação.





Para voltar ao editor de *Python*, onde poderá alterar os dados de entrada, por exemplo, há mais do que um procedimento à escolha, baseados no botão do touchpad. Pode fazer deslocar o cursor com o dedo até o sobrepor ao retângulo com a designação da página, 1.1, neste caso, e premir o touchpad na parte central ( ). Pode também utilizar os botões laterais do touchpad após premir a tecla ctrl. Neste caso, ao premir o botão lateral esquerdo, vai para a página anterior, a do editor. Pode voltar à página de *Shell Python* utilizando o mesmo tipo de procedimento, mas agora com o botão lateral direito.
Na parte superior do ecrã apenas se pode observar a designação e 3 páginas consecutivas, pelo que se o documento tiver mais páginas terá de conjugar os dois procedimentos referidos ou simplesmente o que recorre às teclas laterais do touchpad.

Algumas ideias sobre programação, relacionadas com o contexto