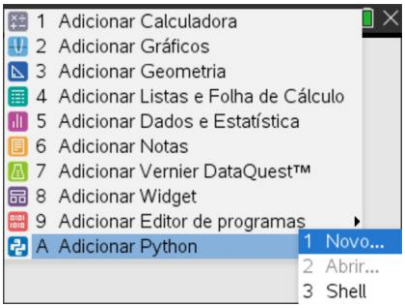
```
x=int(input("Indica o 1.° termo da sucessão: "))
seq=[]
def collatz_sequencia(x):
  seq = [x]
  if x < 1:
    return []
  while x > 1:
    if x % 2 == 0:
      x = x / 2
    else:
      x = 3 * x + 1
    seq.append(int(x))
  return seq
print(collatz_sequencia(x))
```

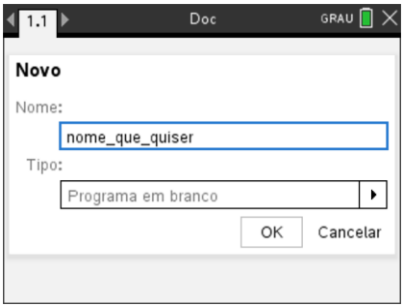
Editar Python na TI-nspire CX II-T

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

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

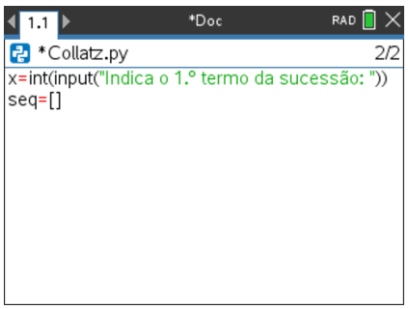
Coloque um nome à sua escolha, de seguida, prime em **OK**.
Abre-se uma página vazia, que é o editor de *Python* da calculadora/tecnologia TI-Nspire CX II-T, onde deve escrever o código.





1. Dado um número inteiro positivo, se for par calcula-se a sua metade, caso contrário, multiplica-se por 3 e adiciona-se uma unidade. Ao resultado obtido continua a aplicar-se a regra até obter 1. Escrever a sequência de valores.

I. Embora não esteja provado, nunca se deixou de obter 1. A este resultado chama-se conjetura de Collatz. Esta situação ocorre na página 21 como um exemplo de um programa em Python para simular a sequência de Collatz obtida a partir de um número dado (previamente fixado)
Embora seja um bom princípio importar o módulo de matemática, as instruções a utilizar funcionam sem o módulo, pelo que se ultrapassará essa parte.



A 1ª linha do programa serve para que o utilizador introduza o número inicial quando o programa for executado.

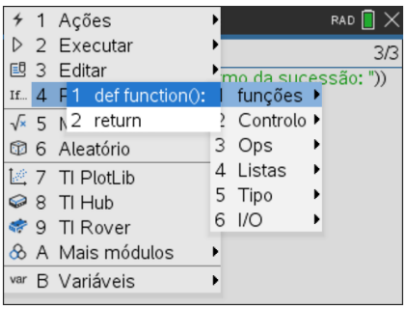
x=int(input("Indica o 1.º termo da sucessão: "))

int e input podem ser escritas com o teclado ou obtidas em menu 4 5 1, para "int" e menu 4 6 2 para "input"

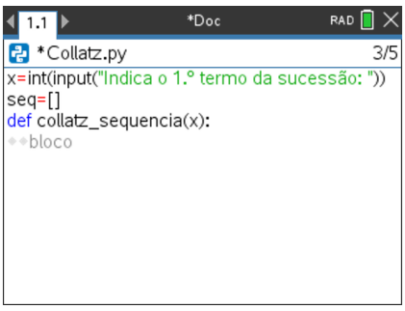
A 2ª linha (**seq=[]**) serve para abrir uma lista vazia onde será colocado o número inicial e os seguintes, terminando com 1 (a sequência).

II. Para continuar a escrita do programa, define-se uma função que, a partir do valor inicial, introduza os valores adequados na lista **seq** até que seja colocado o 1.
Para definir uma função, caso não se escreva com o teclado, segue-se a sequência

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

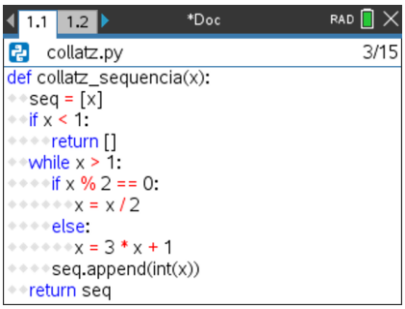


No espaço para o nome da função escreva-se ***collatz_sequencia*** e no argumemto ***x***, que é o valor inicial introduzido pelo utilizador.

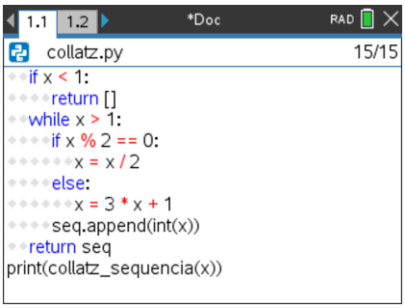


No bloco serão colocadas as instruções. Começa por se devolver uma lista vazia se o número introduzido for inferior a 1. Se tal não acontecer., será colocada uma conjugação de estruturas condicionais para analisar a paridade e efetuar a operação de acordo com a mesma e o resultado será adicionado à lista **seq**.

```
◆◆seq = [x]  # coloca o valor introduzido pelo utilizador na lista seq.
◆◆if x < 1:
◆◆◆◆return [] # devolve uma lista vazia se o valor introduzido for inferior a 1.
◆◆while x > 1: # Enquanto os sucessivos valores, inicial e calculados, forem superiores a 1....
◆◆◆◆if x % 2 == 0: # Se x for par (o resto da divisão por 2 for 0)
◆◆◆◆◆◆x = x / 2 # x passa a valer metade
◆◆◆◆else: # se x não for par
◆◆◆◆◆◆x = 3 * x + 1 # x passa a valer o triplo adicionado de 1
◆◆◆◆seq.append(int(x)) # o valor calculado, num ou noutro caso, é adicionado à lista seq.
◆◆return seq # Se deixou de se verificar que x>1, então é devolvida a sequência construída.
```

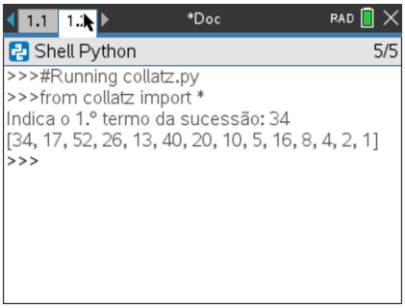


Para terminar falta apenas apresentar a sequência, o que sucede com a instrução **print(collatz_sequencia(x))** fora da função.



Nota: Para executar o programa pode utilizar-se uma instrução do menu (menu 2 1), mas é claramente mais simples utilizar um atalho, uma combinação de teclas (ctrl + R).

O ecrã ao lado revela a execução para x=34, por exemplo. Experimente outras opções e tente obter sequências cada vez maiores.



Algumas ideias sobre programação, relacionadas com o contexto