

```
from random import *
n=int(input("Quantos lançamentos queres simular?"))
f1=0
f2=0
f3=0
f4=0
f5=0
f6=0
for i in range(0,n):
    r=randint(1,6)
    if r==1:
        f1=f1+1
    elif r==2:
        f2=f2+1
    elif r==3:
        f3=f3+1
    elif r==4:
        f4=f4+1
    elif r==5:
        f5=f5+1
    elif r==6:
        f6=f6+1
print("Freq. relativa da saída da face 1 = "+str(f1/n))
print("Freq. relativa da saída da face 2 = "+str(f2/n))
print("Freq. relativa da saída da face 3 = "+str(f3/n))
print("Freq. relativa da saída da face 4 = "+str(f4/n))
print("Freq. relativa da saída da face 5 = "+str(f5/n))
print("Freq. relativa da saída da face 6 = "+str(f6/n))
```

Editar Python na TI-nspire CX II-T

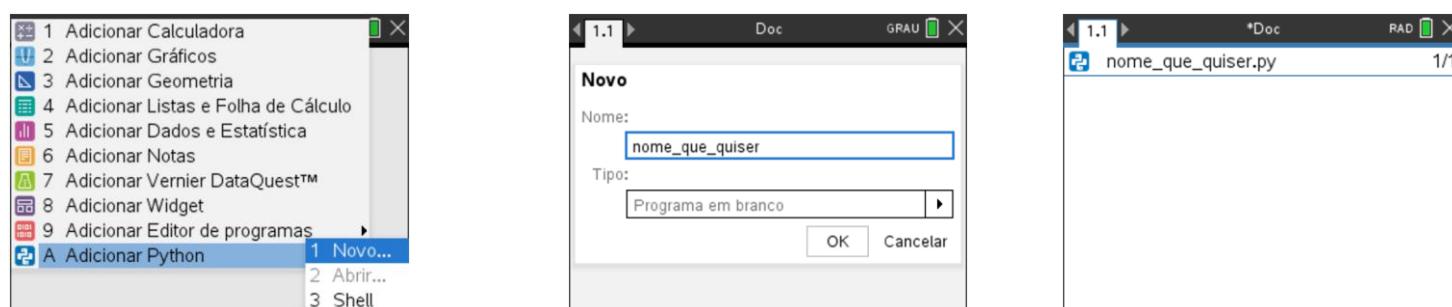
Ligue a sua calculadora e crie um novo documento.

Escolha uma página de *Python*:

  Adicionar Python →  Novo.

Coloque um nome à sua escolha, de seguida, prime em .

Abre-se uma página vazia, que é o editor de *Python* da calculadora/tecnologia TI-Nspire CX II-T, onde deve escrever o código.



I. Como simular o lançamento de um dado n vezes e calcular a frequência relativa de cada uma das seis faces ?

I. Para simular o lançamento de um dado é necessário gerar números aleatórios.

Inicialmente, é preciso, no programa, importar a biblioteca que contem as ferramentas que são necessárias.

Pode-se, assim, escrever com o teclado ou obter no menu.

[menu] → [6] Aleatório → [1] from random import *

The screenshot shows the TI-Nspire CX CAS software interface. A context menu is open with the following items: 1 Ações, 2 Executar, 3 Editar, 4 Planos integrados, 5 Matemática, 6 Aleatório (which is selected and highlighted in blue), 7 TI PlotLib, 8 TI Hub, 9 TI Rover, and A Mais módulos. Below the menu, a code completion dropdown is visible with the following options: 1 from random import *, 2 random(), 3 uniform(min, max), 4 randint(min, max), 5 choice(sequence), 6 randrange(start, stop, step), and 7 seed().

II. O número de lançamentos que serão realizados é da escolha do utilizador.

Pretende-se, assim, fazer um programa que, depois de executado, pergunte ao utilizador o número (inteiro) de vezes que o dado seja lançado.

A função que permite receber um dado por parte do utilizador é **input()** e para que este seja atribuído numa variável, à qual deve ser dado um nome, tem de ser identificado que se trata de um valor numérico inteiro, com **int()**.

n=int(input("Quantos lançamentos queres simular?"))

Para obter a função **input()**, pode escrever com o teclado ou, a partir do menu:

[menu] [4] Planos integrados → [6] I/O → [2] input()

(Para escrever palavras, letras ou texto, que não sejam variáveis, tem de colocar aspas " ".)

III. À medida que são efetuadas as simulações, pretende-se que o programa, no final, devolva as frequências relativas da ocorrência de cada uma das faces do dado.

Para tal, o programa necessita de ir registando o número de vezes que cada uma das faces do dado sai, ou seja, que vá registando as respetivas frequências absolutas.

Assim, tem-se que criar variáveis “contadores”, **f1**, **f2**, **f3**, **f4**, **f5** e **f6**, que contam quantas vezes a respetiva face do dado saiu.

No inicio todos os “contadores” começam com 0. Tem-se, então, as seguintes linhas de código:

f1=0

f2=0

f3=0

f4=0

f5=0

f6=0

The screenshot shows the TI-Nspire CX CAS software interface with the code **n=int(input("Quantos lançamentos queres simular?"))** entered in the editor. The status bar at the top right indicates 4/38 lines of code.

The screenshot shows the TI-Nspire CX CAS software interface with the full code:
from random import *
n=int(input("Quantos lançamentos queres simular?"))
f1=0
f2=0
f3=0
f4=0
f5=0
f6=0



IV. A simulação em questão é realizada tantas vezes quantas o número de vezes que o utilizador quis, ou seja. n vezes.

Perante esta natureza repetitiva da experiência do lançamento do dado, pode-se, em usar um ciclo de repetição **for**, que permite simular os n lançamentos.

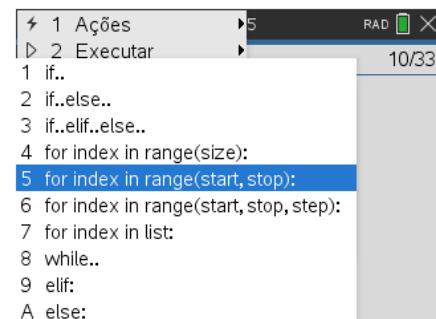
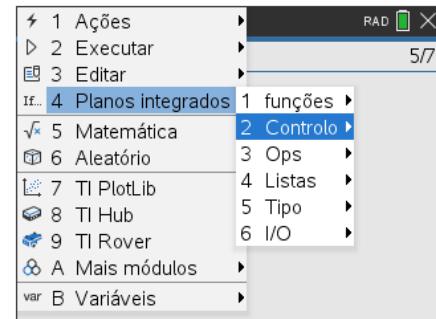
```
for i in range(0,n):
```

◆◆

(Note que a variável i inicia em 0 e incrementa até ao valor anterior ao que está entre parênteses)

Para obter estas linhas de código pode utilizar o teclado, não esquecendo dos “:” e também da indentação “◆◆”, para o que o ciclo de repetição **for** se aplique apenas ao que estiver indentado. Pode também recorrer ao menu para obter as linhas de código, a completar com os elementos específicos.

menu **4 Planos integrados** → **2 Controlo** → **5 for index in range(start,stop):**



V. Para gerar as faces do dado, recorre-se à função **randint** que permite gerar números inteiros aleatórios.

Para obter esta função, pode utilizar o teclado ou recorrendo ao menu:

menu **6 Aleatório** → **4 randint(min, max)**

De modo a representarmos cada face do dado que sai, precisamos de gerar números inteiros aleatórios de 1 a 6. Assim, insere-se, no ciclo de repetição anterior:

```
r=randint(1,6)
```

Onde r é o valor gerado em cada simulação.

Conforme o número que sair em cada simulação, cada variável “contadora” correspondente será adicionada de 1, ou seja:

Se $r = 1$ então

f1 passará a valer $f1 + 1$

Senão, se $r = 2$ então

f2 passará a valer $f2 + 1$

Senão se $r = 3$ então

f3 passará a valer $f3 + 1$

Senão se $r = 4$ então

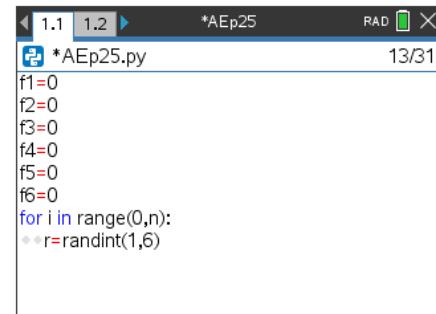
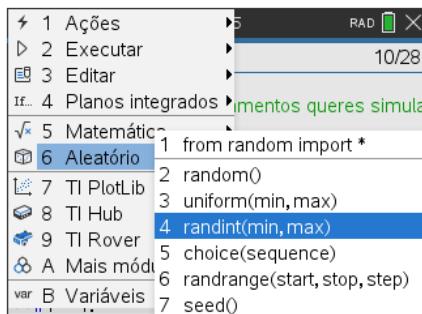
f4 passará a valer $f4 + 1$

Senão se $r = 5$ então

f5 passará a valer $f5 + 1$

Senão se $r = 6$ então

f6 passará a valer $f6 + 1$



Na linguagem **Python**, a linguagem natural que se encontra à esquerda corresponde às seguintes linhas de código:

```
if r==1:  
    f1==f1+1  
elif r==2:  
    f2==f2+1  
elif r==3:  
    f3==f3+1  
elif r==4:  
    f4==f4+1  
elif r==5:  
    f5==f5+1  
elif r==6:  
    f6==f6+1
```

Neste caso, é necessário colocar uma estrutura condicional para que, em cada simulação, o programa atualize as respetivas frequências absolutas.



Para se obter a estrutura condicional, pode se utilizar o teclado, não esquecendo os “:” e também a indentação “◆◆”, para o que as condições se linhas de código, aqui com vantagem pela relativa complexidade da estrutura, a completar depois com os elementos específicos.

menu [4] Planos integrados → [2] Controlo → [3] if...elif...else...

apliquem apenas ao que estiver identado. Pode-se também recorrer ao menu para obter as linhas de código, a ccompletar com elementos específicos

Note que, das três estruturas de condição da lista pendente (iniciadas por **if**), foi escolhida esta (**if...elif...else...**) porque havia necessidade de considerar mais do que uma condições.

Esta estrutura funciona da seguinte forma: se não se verifica a condição inicial, a seguir a **if**, então passa-se para uma outra condição, a atuar sobre o que não se verificou antes. A condição será precedida de **elif** (o número de vezes necessário), ou **else** quando não há mais condições e não se verificou qualquer das anteriores.

- VI.** Em cada uma das simulações, um novo número é gerado e, consequentemente, as frequências absolutas vão se alterar. Portanto, as linhas de código que permitem a contabilização do número de faces que saíram respetivamente, têm de se encontrar dentro do ciclo de repetição (identadas), como se encontra ilustrado nas imagens ao lado.

```

1.1 1.2 *AEp25 RAD 9/28
*AEp25.py
for i in range(0,n):
    r=randint(1,6)
    if r==1:
        f1=f1+1
    elif r==2:
        f2=f2+1
    elif r==3:
        f3=f3+1
    elif r==4:
        f4=f4+1
    elif r==5:
        f5=f5+1
    elif r==6:
        f6=f6+1

```

```

1.1 1.2 *AEp25 RAD 21/28
*AEp25.py
if r==1:
    f1=f1+1
elif r==2:
    f2=f2+1
elif r==3:
    f3=f3+1
elif r==4:
    f4=f4+1
elif r==5:
    f5=f5+1
elif r==6:
    f6=f6+1

```

- VII.** Cada frequência relativa é calculada do seguinte modo:

$$\text{Freq. relativa (face } x) = \frac{\text{Nº de vezes que a face } x \text{ saiu}}{\text{Nº total de lançamentos (n)}}$$

De maneira a concluir o programa é necessário colocar as instruções que permitem a observação ou a impressão da informação pretendida, ou seja, das frequências relativas de cada face do dado, na página Shell Python, que se abre automaticamente logo após a execução do programa.

Deste modo, insere-se as seguintes linhas de código, cada uma das linhas para cada face:

```

print("Freq. relativa da saída da face 1 = "+ str(f1/n))
print("Freq. relativa da saída da face 2 = "+ str(f2/n))
print("Freq. relativa da saída da face 3 = "+ str(f3/n))
print("Freq. relativa da saída da face 4 = "+ str(f4/n))
print("Freq. relativa da saída da face 5 = "+ str(f5/n))
print("Freq. relativa da saída da face 6 = "+ str(f6/n))

```

Como se pretende obter o valor decimal das frequências relativas, tem-se de converter o valor da frequência relativa da face do dado x , f_x/n , para *string* – $\text{str}(f1/n)$ – e, depois, concatenar à *string* “Freq. relativa da saída da face 1 =”, colocando-se + entre as *strings*.

```

1.1 1.2 *AEp25 RAD 23/28
*AEp25.py
f4=f4+1
elif r==5:
    f5=f5+1
elif r==6:
    f6=f6+1
print("Freq. relativa da saída da face 1 = "+str(f1/n))
print("Freq. relativa da saída da face 2 = "+str(f2/n))
print("Freq. relativa da saída da face 3 = "+str(f3/n))
print("Freq. relativa da saída da face 4 = "+str(f4/n))
print("Freq. relativa da saída da face 5 = "+str(f5/n))
print("Freq. relativa da saída da face 6 = "+str(f6/n))

```

VIII. 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**).

O resultado aparece numa nova página destinada a mostrar o resultado da execução do programa, **Shell Python**, na qual também e podem fazer operações e programas, mas que não permanecerão gravados após o fecho da aplicação.

The screenshot shows a software interface titled "Shell Python". At the top, there are tabs for "1.1" (selected) and "1.2", along with a file path "*AEp25" and a battery icon. The main window displays the following text:
>>>#Running AEp25.py
>>>from AEp25 import *\nQuantos lançamentos queres simular?10000\nFreq. relativa da saída da face 1 = 0.1689\nFreq. relativa da saída da face 2 = 0.1657\nFreq. relativa da saída da face 3 = 0.1712\nFreq. relativa da saída da face 4 = 0.1671\nFreq. relativa da saída da face 5 = 0.1594\nFreq. relativa da saída da face 6 = 0.1677
>>>

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.

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

