



```
from random import *
n=int(input("Quantos lançamentos queres simular?"))
f=[0 for k in range(6)]
for i in range(n):
    r=randint(1,6)
    f[r-1]=f[r-1]+1
for k in range(6):
    print("Freq. relativa da saída da face",k+1,"=",f[k]/n)
```

Editar Python na TI-nspire CX II-T

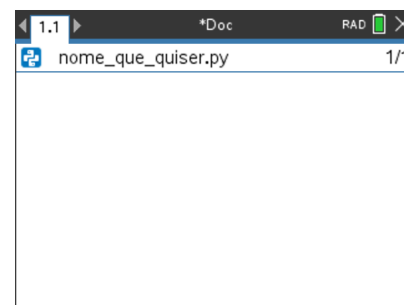
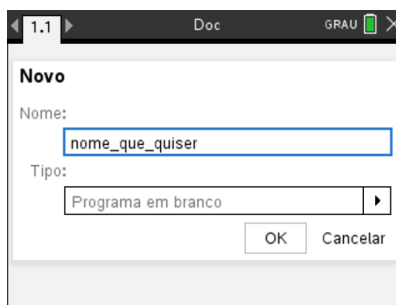
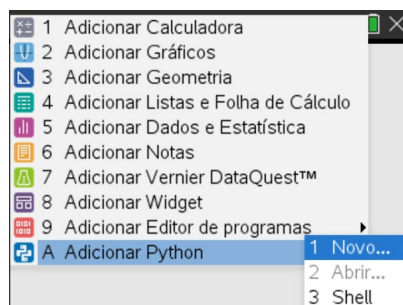
Ligue a sua calculadora e crie um novo documento.

Escolha uma página de *Python*:

[menu] [A] Adicionar Python → [1] Novo.

Coloque um nome à sua escolha, de seguida, prime em **[OK]**.

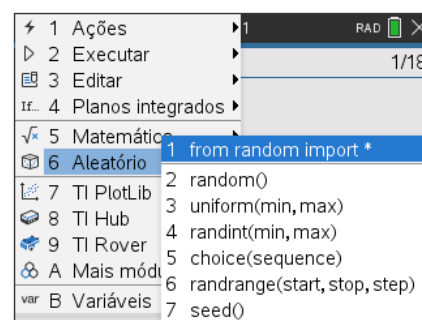
Abre-se uma página vazia, que é o editor de *Python* da calculadora/tecnologia TI-Nspire CX II-T, onde deve escrever o código.



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 *



- 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 será lançado.

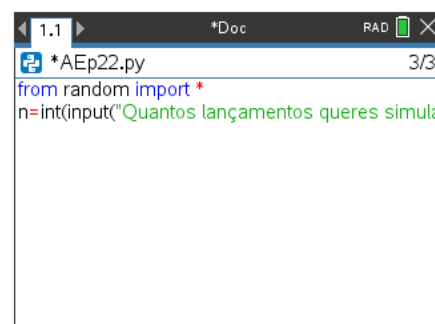
A função que permite receber um dado alfanumérico por parte do utilizador é **input()** e para que este seja atribuído numa variável, à qual deve ser dado um nome, tem de ser identificado que se trata de um valor numérico inteiro, com **int()**.

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. Pretende-se que o programa, no final, devolva as frequências relativas da ocorrência de cada uma das faces do dado, depois de ser feita uma recolha em função do número de experiências definido.

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, têm de ser criados “contadores”, que registam quantas vezes a respetiva face (f1, f2, f3, f4, f5 e f6) do dado saiu. No início todos os “contadores” começam com 0. Poderiam ser colocadas as seguintes seis linhas de código:

f1=0

f2=0

f3=0

f4=0

f5=0

f6=0

Mas por uma questão de eficiência pode utilizar-se apenas uma lista em que cada um dos seis elementos sejam os f# da listagem anterior. Vamos criar a lista, designando-a por **f** (variável de lista), com seis zeros (o mesmo número de faces) para ir contabilizando quantas vezes saiu cada face do dado.

Ou seja, na referida lista, tem-se que:

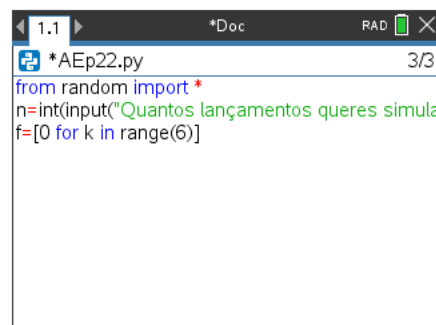
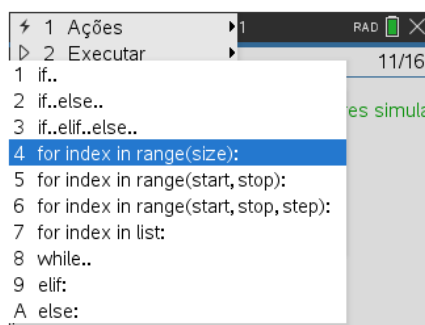
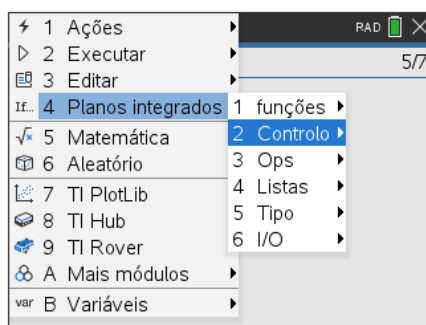
- **f[0]** será o número de vezes que saiu a face 1;
- **f[1]** será o número de vezes que saiu a face 2;
- ...
- **f[5]** será o número de vezes que saiu a face 6.

Assim, para se criar esta lista pode usar-se um ciclo de repetição que permite gerar os seis zeros, elementos iniciais da lista:

f=[0 for i in range(6)]

Para obter esta linha de código pode-se utilizar o teclado. Contudo, pode-se também recorrer ao menu para obter as linhas de código, a completar com os elementos específicos.

menu [4] **Planos integrados** → [2] **Controlo** → [4] **for index in range(size):**



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

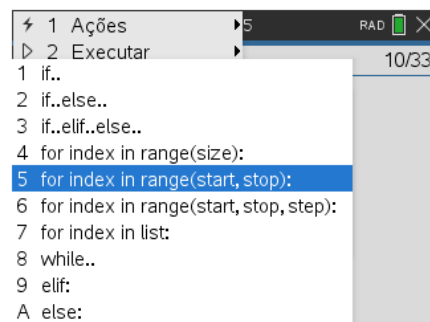
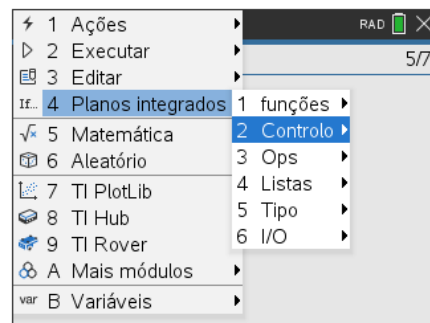
Perante esta natureza repetitiva da experiência do lançamento do dado, pode-se usar um ciclo de repetição **for**, que permite simular os n lançamentos.

for i in range(0,n):

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):**

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



V. Para gerar o número de pintas das 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 saí, 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 “contador” correspondente será adicionado em 1 unidade, ou seja:

Se $r = 1$ então

f[0] passará a valer **f[0] + 1**

Senão, se $r = 2$ então

f[1] passará a valer **f[1] + 1**

Senão se $r = 3$ então

f[2] passará a valer **f[2] + 1**

Senão se $r = 4$ então

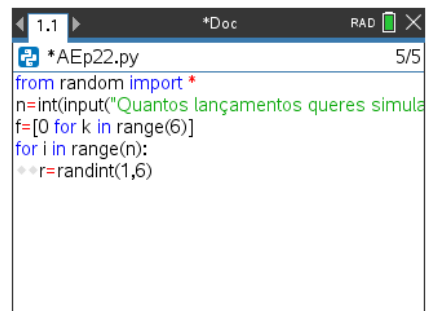
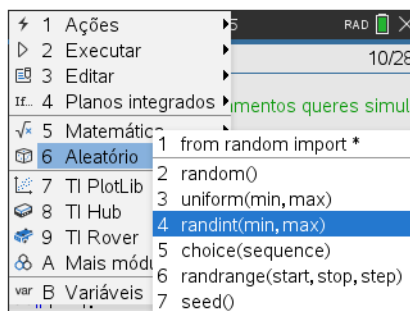
f[3] passará a valer **f[3] + 1**

Senão se $r = 5$ então

f[4] passará a valer **f[4] + 1**

Senão se $r = 6$ então

f[5] passará a valer **f[5] + 1**



Com os “contadores” inseridos numa lista, estas linhas de código podem se reduzir a uma simples linha de código.

Para tal, é preciso ter atenção que os “contadores” estão numa lista e, em *Python*, as listas começam no índice 0. Portanto, perante o número saído, o incremento será feito no elemento da lista adequado, pelo que depois se terá de retirar, ao valor gerado, uma unidade, ou seja: **r-1**

Por exemplo, se $r = 2$ (saiu a face 2 do dado), adiciona-se 1 unidade em **f[1]** e assim sucessivamente.

VI. Com já foi referido anteriormente, em cada uma das simulações, um novo número natural aleatório de 1 a 6 é gerado e, consequentemente, as frequências absolutas vão se alterar. Assim, a linha de código que permite a contínua contabilização do número de faces que saíram respetivamente, é a seguinte e têm de se encontrar dentro do ciclo de repetição:

f[r-1]=f[r-1]+1

VII. Cada frequência relativa é calculada do seguinte modo:

$$\text{Freq. relativa (face } x) = \frac{\text{N}^{\circ} \text{ de vezes que a face } x \text{ saiu}}{\text{N}^{\circ} \text{ total de lançamentos (n)}}$$

No caso em questão, para se calcular a frequência relativa relativa a uma certa face, tem-se de “chamar” o valor da lista **f** respetivo.

Por exemplo, pretende-se calcular a frequência relativa da face 3, então será **f[2]/n**.

Portanto, tendo em isto em consideração, para 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.

Para além disto, de modo a não se ter que escrever seis linhas de código respetivas às seis faces, utiliza-se, novamente, um ciclo de repetição:

for k in range(6):
 ♦♦ **print(("Freq. relativa da saída da face", k+1, " = ", f[k]/n))**

É de notar que **k+1** serve para mostrar a face correta do dado (1 a 6), já que os índices da lista, **k**, vão de 0 a 5.

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.

```
1.1 *Doc RAD 6/6
*AEp22.py
from random import *
n=int(input("Quantos lançamentos queres simula
f=[0 for k in range(6)]
for i in range(n):
    r=randint(1,6)
    f[r-1]=f[r-1]+1
```

```
1.1 *Doc RAD 8/8
*AEp22.py
from random import *
n=int(input("Quantos lançamentos queres simula
f=[0 for k in range(6)]
for i in range(n):
    r=randint(1,6)
    f[r-1]=f[r-1]+1
for k in range(6):
    print("Freq. relativa da saída da face",k+1,"=",
```

```
1.1 1.2 *Doc RAD 10/10
Shell Python
>>>#Running AEp22.py
>>>from AEp22 import *
Quantos lançamentos queres simular?1000
Freq. relativa da saída da face 1 = 0.172
Freq. relativa da saída da face 2 = 0.152
Freq. relativa da saída da face 3 = 0.161
Freq. relativa da saída da face 4 = 0.176
Freq. relativa da saída da face 5 = 0.157
Freq. relativa da saída da face 6 = 0.182
>>>
```

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

