



Unidade 3: Iniciação à programação em Python

Lição 2: O ciclo FOR

Nesta segunda lição da Unidade 3, descobrirá como repetir um processo ou um conjunto de instruções utilizando o ciclo **FOR**.

Objetivos:

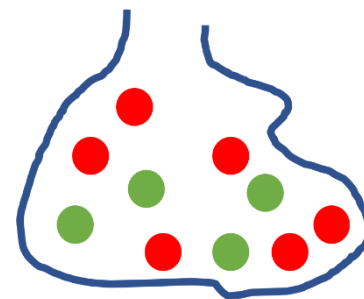
- Aplicar uma função.
- Descobrir como implementar o ciclo **FOR**
- Utilizar o ciclo **FOR** em exemplos simples.

Simular uma Experiência Aleatória

Um saco opaco contém seis bolas vermelhas e quatro verdes. Retira-se, ao acaso, uma bola do saco e regista-se a sua cor, de seguida volta-se a colocar a bola no saco.

Crie um programa com uma função **cor()** que simule esta experiência aleatória de variável X.

- Que valores pode tomar a variável X?
- Pretende-se escrever um programa que, através de uma condição, distinga as bolas vermelhas das bolas verdes.



- Comece um novo programa, com o nome "**EXPERIENCIA**".
- Como irá trabalhar com números aleatórios, é necessário carregar a biblioteca "random". Para isso, pressionar a tecla **[menu]**, depois **6: Aleatório**.
- Insira o programa constante no ecrã ao lado no editor de TY-Python, tendo em atenção à necessidade de respeitar a indentação.

```
1.1 *Doc RAD 8/8
*experiencia.py
from random import *
def cor():
    x=randint(1,10)
    if x<=6:
        c="vermelho"
    else:
        c="verde"
    return c
```

- Executar o programa e, no interpretador (Shell) exibir o resultado da função **cor()**.
- Repita a execução da função **cor()**, poderá escrever o nome e pressionar **[enter]** ou clicar na tecla **[var]**, e analise os resultados.

```
1.1 1.2 *Doc RAD 9/9
Shell Python
>>>#Running experiencia.py
>>>from experiencia import *
>>>cor()
'verde'
>>>cor()
'verde'
>>>cor()
'vermelho'
>>>
```

SUGESTÃO:

Este programa pode ser alterado para se poder aplicar a qualquer número de bolas. Neste caso, pode definir-se a função **cor(n,a)** onde os argumentos representam: **n** é o número de bolas e **a** o número de bolas vermelhas.



APLICAÇÃO DAS APRENDIZAGENS:

Amostragem e toma de decisão

Fez-se uma sondagem acerca de um novo espetáculo proposto por um artista. O estudo, realizado numa grande cidade, revelou que dois terços das pessoas que viram o espetáculo gostaram. O agente do artista acredita que toda a população portuguesa pensará da mesma forma. Para verificar o que pensou, solicitou um estudo a uma empresa de sondagens.



Estudo da População

Para realizar o estudo, o estatístico da empresa deve criar uma função que simule a situação.

O vosso trabalho consiste em criar esta função, respeitando as seguintes regras:

- O espetáculo agradou, com uma probabilidade $p = \frac{2}{3}$.
- O espetáculo não agradou, com uma probabilidade $p = \frac{1}{3}$.

1. Comece um novo programa, com o nome **SONDAGEM**.
2. Escreva a função no editor do TI-Python e teste-a várias vezes pressionando **ctrl** **R** e depois digite na Shell o nome da função, sem argumentos, **resposta()** e pressione **enter**.

```
*sondagem.py 8/8
from random import *
def pergunta():
    s=randint(0,2)
    if s==0 or s==1:
        resposta=1
    else:
        resposta=0
    return resposta
```

DICA:

Pode utilizar a tecla **var** e seleccionar a função ou utilizar a tecla de seta para cima, **▲**, até seleccionar a última instrução e clicar **enter** para repetir a execução (pode ser mais rápido do que digitar o nome da função...).

Simulação duma amostra de dimensão n .

A empresa de sondagens pretende simular amostras de dimensões variadas. Portanto, deve criar no editor de programas uma função **experiencia(n)** que execute a função **pergunta()** n vezes, isto é, gere uma amostra de dimensão n .

- Inicie a função com uma instrução que crie uma lista vazia, lista **L**.
- Preencha esta lista utilizando a função **pergunta()** e um ciclo **For**.

```
*sondagem.py 10/13
s=randint(0,2)
if s==0 or s==1:
    resposta=1
else:
    resposta=0
return resposta

def experiencia(n):
    L=[]
    L=[pergunta() for i in range(n)]
    return L
```

SUGESTÃO:

A linguagem Python permite que se utilize uma função para preencher uma lista incrementada por um ciclo **For**, como argumento da lista. Processo análogo ao da função **seq()** usada, nas restantes aplicações da TI-Nspire CX II, para gerar listas.

Este exemplo pode ser feito sem listas. Nesse caso, o programa deve ser ligeiramente alterado para substituir as instruções relativas às listas por ciclos com incremento de uma variável.

- Testar a função **experiencia(n)** com uma amostra de dimensão 20, executando o programa e escrevendo **experiencia(20)** no interpretador e pressionando a tecla **enter**.

```

Shell Python 6/6
>>>#Running sondagem.py
>>>from sondagem import *
>>>experiencia(20)
[1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0]
>>>
    
```

Validação da experiência.

A empresa de sondagens deseja determinar a percentagem de amostras cuja percentagem de pessoas que gostaram do espetáculo se enquadram no intervalo de confiança a 95% com $p = \frac{2}{3}$.

Dando seguimento ao programa anterior, deverá criar duas novas funções:

- freq_amostra(n)** para calcular automaticamente a frequência de respostas iguais a 1, observadas na amostra L de dimensão n.
- interv_prev(da,na)** para determinar, em na amostras, o número de amostras (Nf) em que a frequência pertence ao intervalo de confiança de 95%, considerando amostras de dimensão da.

ALGORITMO:

Função `interv_prev(dim_amostra, num_amostra)`

$Nf \leftarrow 0$

Para i de 1 a num_amostra faz

$f \leftarrow \text{freq_amostra}(\text{dim_amostra})$

Se f pertence a $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}} \right]$ então

$Nf \leftarrow Nf + 1$

FimSe

FimPara

RECORDE QUE:

Para uma dada característica estatística, neste caso frequência ou proporção (\hat{p}), dos elementos de uma amostra, ao seu valor na população designa-se por parâmetro e representa-se por p .

Prova-se que para $n \geq 25$ e $0.2 \leq p \leq 0.8$, a frequência da característica na amostra de dimensão n pertence ao intervalo $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}} \right]$ em 95% dos casos. Este intervalo designa-se “Intervalo de Confiança a 95%”.

A partir do algoritmo da função **interv_prev()**, escrito em linguagem natural, crie, no programa já iniciado no editor do TI Python, a função **interv_prev()**.

Observe o ecrã ao lado para se orientar.

Execute o programa, atalho **ctrl** + **R**.

- No interpretador, pressione a tecla **var** para chamar a função **interv_prev**.
- Teste o programa diversas vezes para amostras de dimensão 100.
- Deduza para dado um exemplo, qual o intervalo de confiança a 95%. Para tal pode seguir a estratégia de fixar a dimensão de cada amostra (**da**) e variar o número de amostras (**na**).
- Este estudo estatístico coloca em causa a afirmação do agente do artista?

```

sondagem.py 14/22
L=[pergunta() for i in range(n)]
return sum(L)/len(L)
|
from math import sqrt
def interv_prev(da,na):
    Nf=0
    for i in range(na):
        f=experiencia(da)
        if 2/3-1/sqrt(da)<=f<=2/3+1/sqrt(da):
            Nf+=1
    return f
    
```

```

Shell Python 48/48
0,64
>>>
>>>interv_prev(100,100)
0,67
>>>interv_prev(100,100)
0.6899999999999999
>>>interv_prev(100,100)
0.6899999999999999
>>>interv_prev(100,100)
0,75
>>>|
    
```

