

# Computação I - Python

## Aula 9: Laços Aninhados e Matrizes

### Construção de uma tabuada da multiplicação utilizando laços aninhados

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



## Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

- Há diversas lógicas distintas para se escrever uma tabuada no papel. Vamos aqui montar a tabuada da multiplicação da seguinte maneira:

Considerando cada linha da tabuada no formato  $A \times B = A*B$ :

- 1 Escrevemos primeiramente a tabuada de 1, completa:  
Ex: (  $1 \times 1 = 1$ ,  $1 \times 2 = 2$ , . . . ,  $1 \times 9 = 9$  )
- 2 Terminando a tabuada de 1, escrevemos a tabuada de 2:  
Ex: (  $2 \times 1 = 2$ ,  $2 \times 2 = 4$ , . . . ,  $2 \times 9 = 18$  )
- 3 Repetimos o passo 2 acima até escrevermos a tabuada de 9:  
Ex: (  $9 \times 1 = 9$ ,  $9 \times 2 = 18$ , . . . ,  $9 \times 9 = 81$  )

Observe que podemos reescrever os passos acima de maneira mais genérica.

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

- Há diversas lógicas distintas para se escrever uma tabuada no papel. Vamos aqui montar a tabuada da multiplicação da seguinte maneira:

Considerando cada linha da tabuada no formato  $A \times B = A*B$ :

- 1 Fixamos  $A = 1$ , e iteramos o valor de  $B$ , de 1 a 9.  
Para cada iteração, escrevemos a expressão  $1 \times B = 1*B$ .
- 2 Incrementamos em 1 o valor de  $A$ , ou seja, agora  $A = 2$ .  
Iteramos novamente o valor de  $B$ , de 1 a 9.  
Para cada iteração, escrevemos a expressão  $2 \times B = 2*B$ .
- 3 Repetimos o passo 2 acima até que  $A = 9$ .  
Iteramos novamente o valor de  $B$ , de 1 a 9.  
Para cada iteração, escrevemos a expressão  $9 \times B = 9*B$ .

Ainda é possível reescrever os passos acima de maneira ainda mais sucinta.

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

- Há diversas lógicas distintas para se escrever uma tabuada no papel. Vamos aqui montar a tabuada da multiplicação da seguinte maneira:

Considerando cada linha da tabuada no formato  $A \times B = A*B$ :

- 1 Iteramos o valor de  $A$ , de 1 a 9.  
Para cada iteração de  $A$ , iteramos o valor de  $B$ , de 1 a 9.  
Para cada iteração de  $B$ , escrevemos a expressão  $A \times B = A*B$ .

Podemos implementar essa lógica, em Python utilizando 2 laços aninhados, onde o laço mais externo representará as iterações de  $A$ , e o laço mais interno representará as iterações de  $B$ .

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

- Considerando cada linha da tabuada no formato  $A \times B = A*B$ :
  - ❶ Iteramos o valor de  $A$ , de 1 a 9.
    - Para cada iteração de  $A$ , iteramos o valor de  $B$ , de 1 a 9.
    - Para cada iteração de  $B$ , escrevemos a expressão  $A \times B = A*B$ .

Para implementarmos os 2 laços aninhados, podemos utilizar:

- 2 laços **for**;
- 2 laços **while**;
- 1 laço de cada, na ordem que quisermos.

Para montarmos a tabuada, podemos utilizar, por exemplo, listas ou uma string.

Vejamos diferentes resoluções para o problema:

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

Versão utilizando utilizando 2 laços **for** aninhados e uma lista:

```
def tabuada_v1():  
    '''Gera tabuada no formato A x B = A*B  
    Versao com lista unica  
    None -> list'''  
    tab = []  
    for a in range(1,10):  
        for b in range(1,10):  
            linha = str.format("{} x {} = {}".format(a, b, a*b))  
            list.append(tab, linha)  
    return tab
```

Nesta versão, cada linha da tabuada é dada como uma posição da lista **'tab'**.  
Dessa forma, a lista **'tab'** contém 81 posições.

# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

Versão utilizando utilizando 2 laços **for** aninhados e uma lista de listas:

```
def tabuada_v2():
    '''Gera tabuada no formato A x B = A*B
    Versao com lista de listas
    None -> list'''
    tab = []
    for a in range(1,10):
        numero = []
        for b in range(1,10):
            linha = str.format("{} x {} = {}".format(a, b, a*b))
            list.append(numero, linha)
        list.append(tab, numero)
    return tab
```

Nesta versão, a lista **'tab'** contém 9 posições, e há uma lista em cada uma dessas posições, onde cada lista contém as 9 linhas de uma tabuada. Assim, a tabuada de cada número **'A'** está contida em uma das sublistas de **'tab'**.



# Laços aninhados

Faça uma função que gere uma tabuada da multiplicação, de 1 a 9.

Versão utilizando utilizando 2 laços **for** aninhados e concatenação de strings:

```
def tabuada_v3():  
    '''Gera tabuada no formato A x B = A*B  
    Versao com concatenacao de strings  
    None -> str'''  
    tab = ""  
    for a in range(1,10):  
        for b in range(1,10):  
            tab = tab + str.format("{} x {} = {}".format(a, b, a*b), a, b, a*b)  
            tab = tab + "\n"  
    return tab
```

Nesta versão, todas as 81 linhas da tabuada estão contidas em uma única string, separadas pelo caractere de quebra de linha, '\n'. Cada tabuada de um número 'A' é separada da tabuada do número 'A+1' também pelo caractere '\n', gerando uma linha em branco entre cada tabuada.

## Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

## Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

# Computação I - Python

## Aula 9: Laços Aninhados e Matrizes

### Construção de uma tabuada da multiplicação utilizando laços aninhados

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>

