



Engenharia de Software

Computacional thinking with Python

Prof. Dr. Francisco Elânio

Objetivos

- ✓ Revisão sobre min, max, median, mean, range, len
- ✓ Iteração com mais de duas variáveis usando for
- ✓ Exemplos
- ✓ Exercícios
- ✓ Explicação do checkpoint 2

Função min, max, range e len

Min e Max

```
numeros = [10, 11, 15, 18, 20]
```

```
print(f"O valor máximo é", max(numeros))  
print(f"O valor mínimo é", min(numeros))
```

Len

```
numeros = [10, 11, 15, 18, 20]
```

```
print(f"A quantidade de elementos da lista  
é", len(numeros))
```

Mean e Median

```
from statistics import mean, median
```

```
numeros = [10, 11, 15, 18, 20]
```

```
media = mean(numeros)  
mediana = median(numeros)
```

```
print(f'Média: {media}')  
print(f'Mediana: {mediana}')
```

Função min, max, range e len

Range

```
range(fim)  
range(inicio, fim)  
range(inicio, fim, passo)
```

Exemplos

```
print(list(range(5)))  
print(list(range(5, 10)))  
print(list(range(0, 10, 2)))
```

Resultado 1 - [0, 1, 2, 3, 4]

Resultado 2 - [5, 6, 7, 8, 9]

Resultado 3 - [0, 2, 4, 6, 8]

Função Soma e Subtração

Max e Min

```
numeros = [10, 11, 15, 18, 20]
```

```
print(f"O valor máximo é", max(numeros))
```

```
print(f"O valor mínimo é", min(numeros))
```

Média e Mediana

```
from statistics import mean, median
```

```
numeros = [10, 11, 15, 18, 20]
```

```
media = mean(numeros)
```

```
mediana = median(numeros)
```

```
print(f'Média: {media}')
```

```
print(f'Mediana: {mediana}')
```


Função Soma e Subtração

Soma

```
lista = [10, 20, 30, 40, 50]  
soma = sum(lista)  
print(soma)
```

Subtração e Reduce

```
from functools import reduce  
from operator import sub  
lista = [10, 20, 30, 40, 50]  
resultado = reduce(sub, lista)  
print(resultado)
```

Soma usando a função add e reduce

```
from functools import reduce  
from operator import add  
  
lista = [10, 20, 30, 40, 50]  
soma = reduce(add, lista)  
print(soma)
```

Operações com elementos específicos

Soma

```
lista = [10, 20, 30, 40, 50]  
lista[0] + lista[2]
```

Subtração

```
lista = [10, 20, 30, 40, 50]  
lista[1] - lista[3]
```

Multiplicação

```
lista = [10, 20, 30, 40, 50]  
lista[0] * lista[3]
```

Divisão

```
lista = [10, 20, 30, 40, 50]  
lista[2] / lista[0]
```

Estrutura for em Python

Iterador: é uma variável nomeada para iterar.

Iterável: é uma sequência ou coleção através da qual a iteração acontece para loops.

Tarefa: essa é a parte em que o código executa uma tarefa ou uma função.

Exemplo

```
for iterador in iteravel:  
    executa tarefa
```


Loop dentro de um Loop

Iteração entre números

```
for i in range(1, 11):
    for j in range(1, 11):
        print(i*j, end=" ")
```

Primeiro loop é iniciado com a variável i que vai de 1 a 10.

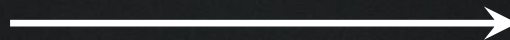
Dentro do primeiro loop, há outro loop for com a variável j que também vai de 1 a 10.

Dentro do segundo loop (for j in range(1, 11:)), a multiplicação de i e j é calculada (i * j).

<i>Iteração</i>	<i>i * j</i>
i = 1	1*1 = 1
	1*2 = 2
	1*3 = 3

i = 2	2*1 = 2
	2*2 = 4
	2*3 = 6

i = 3	3*1 = 3
	3*2 = 6
	3*3 = 9



<i>Resultado</i>										
1	2	3	4	5	6	7	8	9	10	
2	4	6	8	10	12	14	16	18	20	
3	6	9	12	15	18	21	24	27	30	
4	8	12	16	20	24	28	32	36	40	
...										

Loop dentro de um Loop

Iteração entre caracteres e números

```
for letra in 'abc':  
    for numero in '123':  
        print(letra + numero)
```

O primeiro loop itera sobre as letras a, b e c. O segundo loop itera sobre os números 1, 2 e 3.

<i>Primeira Iteração</i>	<i>Segunda Iteração</i>	<i>letra + numero</i>
a	1	a1
	2	a2
	3	a3
b	1	b1
	2	b2
	3	b3
c	1	c1
	2	c2
	3	c3

Loop dentro de um Loop

Usando break

```
for i in range(2, 4):  
  
    for j in range(1, 11):  
        if i==j:  
            break  
        print(i, "*", j, "=", i*j)  
    print()
```

i varia de 2 a 3 e j varia de 1 a 10. Se i for igual a j, o loop interno é interrompido e o programa passa para a próxima iteração do loop externo.

<i>Primeira Iteração</i>	<i>Segunda Iteração</i>	<i>i * j</i>
2	1	2
	2	i == j
3	1	3
	2	6
	3	i == j

Loop dentro de um Loop

Usando continue

```
for i in range(2, 4):  
  
    for j in range(1, 11):  
        if i==j:  
            continue  
        print(i, "*", j, "=", i*j)
```

i varia de 2 a 3 e j varia de 1 a 10. Se i for igual a j, não há iteração entre i e j. Porém, a iteração continua até para i e j.

Primeira Iteração	Segunda Iteração	$i * j$
2	1	2
	2	$i == j$
	3	6
	⋮	-
	10	20
3	1	3
	2	6
	3	$i == j$
	⋮	-
	10	30

For, Continue e Acumulador

```
soma = 0
```

```
for x in [1, 2, 3, 4]:
```

```
    if x == 2:
```

```
        continue
```

```
    soma += x
```

```
print(soma)
```



Os valores de x serão somados a variável soma.

No entanto, para `x == 2`, `continue` não permite que o número 2 seja somado.

For, Break e acumulador

```
soma = 0
```

```
for x in [1, 2, 3, 4]:
```

```
    if x == 2:
```

```
        break
```

```
    soma += x
```

```
print(soma)
```

Os valores de x serão somados a variável soma.

No entanto, quando `x == 2`, `break` não permitirá continuar.

Média Ponderada

Calcular a média ponderada notas, considerando o respectivo peso para cada nota. Por fim, calcule a média ponderada.

```
notas = [7.5, 8.0, 6.5, 9.0]
```

```
pesos = [0.3, 0.2, 0.2, 0.3]
```

```
media_ponderada = sum([nota * peso for nota, peso in zip(notas, pesos)])
```

```
print(f"A média ponderada é {media_ponderada}")
```

Loop dentro de um Loop

Exemplo três listas

```
alunos = ['João', 'Maria', 'Pedro', 'Ana']  
notas = [8.5, 9.0, 7.5, 9.5]  
faltas = [2, 0, 1, 3]
```

```
for i in range(len(alunos)):  
    if notas[i] >= 8.0 and faltas[i] == 0:
```

Itera sobre os índices dos elementos nas listas alunos, notas e faltas.

len(alunos) retorna o número de elementos na lista alunos, que é o número total de alunos.

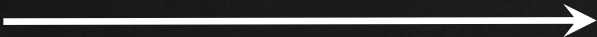
notas[i] >= 8.0: Verifica se a nota do aluno na posição i é maior ou igual a 8.0.

faltas[i] == 0: Verifica se o número de faltas do aluno na posição i é igual a zero.

Loop dentro de um Loop

Exemplo três listas

```
nome = alunos[i]  
nota = notas[i]  
falta = faltas[i]  
    print(f'{nome} foi aprovado com nota {nota}  
    e {falta} falta(s).')
```



Ambas as condições precisam ser verdadeiras para o aluno ser considerado aprovado.

Print os alunos que atendem as condições.

Exercício – Gerador de Senha

Abordagem 1

Crie um algoritmo na qual o usuário possa escolher caracteres e o números que serão utilizados para gerar uma senha.

```
import random
```

```
letras = input("Insira as letras para a senha: ")
```

```
numeros = input("Insira os números para a senha: ")
```

```
senha = ''.join(random.sample(letras + numeros, len(letras + numeros)))
```

```
print(f"A senha gerada é: {senha}")
```


Exercício – Gerador de Senha

Abordagem 2

```
import random
```

```
letras = input("Insira as letras para a senha: ")  
numeros = input("Insira os números para a senha: ")
```

```
senha = ""
```

```
for _ in range(len(letras)):  
    indice_aleatorio = random.randint(0, len(letras)-1)  
    senha += letras[indice_aleatorio]  
    letras = letras[:indice_aleatorio] + letras[indice_aleatorio+1:]
```

```
for _ in range(len(numeros)):  
    indice_aleatorio = random.randint(0, len(numeros)-1)  
    senha += numeros[indice_aleatorio]  
    numeros = numeros[:indice_aleatorio] + numeros[indice_aleatorio+1:]
```

```
senha_embaralhada = ""  
while senha:  
    indice_aleatorio = random.randint(0, len(senha)-1)  
    senha_embaralhada += senha[indice_aleatorio]  
    senha = senha[:indice_aleatorio] + senha[indice_aleatorio+1:]
```

```
print(f"A senha gerada é: {senha_embaralhada}")
```

_: O sublinhado _ indica que a variável não será usada dentro do loop. Não estamos interessados nos valores individuais gerados por `range(len(numeros))`

`random.randint(0, len(numeros)-1)` gera um índice aleatório que está dentro dos limites válidos da string `numeros`.

*O importante é não parar
de questionar (Einstein)*

Referências

ASCENCIO, A. F. G, CAMPOS, E. A. V. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java, 2ª Edição, São Paulo: Pearson 2007.

SALVETTI, Dirceu Douglas; BARBOSA, Lisbete Madson. Algoritmos. São Paulo: Pearson, 2004.