

**targettrust**  
treinamento e tecnologia



# Repetição

---

Às vezes é necessário repetir um conjunto de operações. Como por exemplo, calcular o salário líquido de 100 funcionários a partir de seu salário bruto. Ou calcular os números primos entre um intervalo como por exemplo 2 a 50.

Em Python, os comandos `for` e `while` repetem operações. O comando `do-while` não existe no Python.

# Range e Reversed

---

O comando range é o gerador que nos permite delimitar a quantidade de vezes que queremos repetir o nosso código. Ele possui três parâmetros:

- start (por padrão ele é 0)
- end (não possui padrão, deve ser informado sempre)
- step (por padrão ele é 1)

Exemplo:

range(5) -> 0 1 2 3 4

range(3, 8) -> 3 4 5 6 7

range(0, 11, 2) -> 0 2 4 6 8 10

range(9, 5, -1) -> 9 8 7 6

reversed(range(5)) -> 4 3 2 1 0

reversed(range(3, 8)) -> 7 6 5 4 3

reversed(range(0, 11, 2)) -> 10 8 6 4 2 0

reversed(range(9, 5, -1)) -> 6 7 8 9

# Comando de repetição FOR

```
print('Número pares entre 1 e 50: ')\nfor numero in range(1, 50):\n    if numero % 2 == 0:\n        print(numero, end=' ')
```

```
print('Digite um intervalo (inicio, fim)')
inicio = int(input('Inicio: '))
fim = int(input('Fim: '))

for numero in range(inicio, fim + 1):
    print(str(numero), end=' ')
```

# Comando de Repetição - While

---

O laço de repetição while irá repetir o seu código enquanto a proposição lógica for verdadeira!

Exemplos:

- Enquanto a hora digitada for inválida pergunte para o usuário a hora.
- Enquanto o valor digitado for positivo calcule a raiz quadrada

# Comando de repetição WHILE

```
print('Número impares entre 1 e 50: ')  
numero = 1
```

```
while numero <= 50:  
    if numero % 2 == 1:  
        print(numero, end=' ')  
    numero = numero + 1
```



# Exercício

---

Escreva um algoritmo em Python que conta de 10 a 20 de 2 em 2, e imprime os valores contados



5 min





# Exercício

---

Escreva um algoritmo em Python que exibe na tela todos números primos entre 2 e 50



10 min



# Exercício

---

Escreva um algoritmo em Python que solicite um intervalo de anos e diga todos bissextos nesse intervalo.



10 min

# Strings

---

Uma string em Python é a representação de uma ou mais palavras

- `nome_completo = 'Matheus da Silva Serpa'`
- `mensagem = "Eu gostaria de um copo d'água"`

Em Python, é possível percorrer uma string utilizando um laço for

```
for letra in nome:  
    print(letra)
```

Em Python não existe diferença entre um caractere (char) e um conjunto de caracteres (string)

# Strings - Operações

---

Algumas operações aritméticas podem ser utilizadas com strings

- `nome = 'Matheus'`
- `sobrenome = 'Serpa'`

Concatenação de duas ou mais strings

- `nome_completo = nome + sobrenome`
- `nome_completo = nome + ' ' + sobrenome`

Multiplicação de Strings

- `traco = '-'`
- `separador = traco * 80`

# Tamanho de Strings

---

Dadas algumas strings

- nome = 'Matheus'
- sobrenome = 'Serpa'

Tamanho da string

- len(nome)
- len(nome) / len(sobrenome)
- len('Segunda-feira')

# Métodos com Strings

---

Mais utilizadas

- count, find, format, join, lower, replace, split, strip
- Outras operações
  - <https://docs.python.org/pt-br/3.8/library/stdtypes.html#string-methods>



# Prática

---

Implemente um programa em Python que recebe como entrada uma *string* digitada pelo usuário, e que gera uma nova *string* de saída substituindo todas as vogais (maiúsculas ou minúsculas) pelo caractere `_`. Como saída o programa deve imprimir a *string* gerada.

Veja o exemplo:

```
str = "Eu quero aprender Python"
```

```
Saída = "__ q__r__pr_nd_r Pyth_n"
```



10 min