

PYTHON PARA PENTEST

Aula 2 - Trabalhando com Condições e Repetições
(If, Else, Elif, While e For)





— Trabalhando com Condições

- Python fornece estruturas de controle de fluxo, como condições (if, elif, else) e laços de repetição (for, while), que permitem que você tome decisões e repita tarefas com base em certas condições.

A declaração **if** em Python é usada para executar um bloco de código **se uma condição especificada for verdadeira**.

```
aula = 2
```

```
se aula == 2
```

```
faça
```



— Trabalhando com Condições

- Mas e se a condição for **Falsa**? O bloco de código será ignorado.
- O **else** é usado em conjunto com a declaração **if** para executar um outro bloco de código quando a condição especificada no **if** for avaliada como **falsa**.

```
porta = 443

se porta == 80
    imprima porta padrão HTTP

caso contrário (se porta != 80)
    imprima porta padrão HTTPS
```

— Trabalhando com Condições

- If aninhado

Um **if aninhado** ocorre quando você coloca uma estrutura **if dentro de outra estrutura if**. Isso permite que você verifique condições adicionais dentro de um bloco de código já condicionado por outro **if**.

```
porta = open
escopo = yes

se porta open
se estiver no escopo
imprima a porta está aberta e alvo no escopo
    caso contrário (escopo != yes)
        imprima a porta está aberta, mas o alvo está fora do escopo
caso contrario (porta != open)
imprima a porta está fechada/filtrada
```



— Trabalhando com Condições

- Elife

A estrutura **elif** é usada em Python como uma alternativa ao uso de **múltiplos if aninhados** quando você precisa verificar várias condições em sequência. O **elif** permite testar condições adicionais se as condições anteriores forem avaliadas como falsas.

Em resumo é um **Switch Case**



— Trabalhando com Repetições

- While

O **while** é usado para repetir um bloco de código enquanto uma condição especificada for **verdadeira**.

```
enquanto 1 == 1  
faça
```



— Trabalhando com Repetições

- For

O **for** permite que você execute um bloco de código para cada item em uma sequência (lista, tupla, string).

```
lista = [nome1, nome2, nome3...]
```

```
para cada nome em lista  
faça
```

Muito obrigado por assistir até aqui!

Se você gostou, continue acompanhando o meu conteúdo.

Curta, inscreva-se e compartilhe para ajudar a divulgar o curso e o canal. Até o próximo vídeo!

- LinkedIn: <https://www.linkedin.com/in/elizeudasdores/>
- E-mail: elizeuopain@proton.me
- Github: <https://github.com/ELIZEUOPAIN>

Proxima Aula...

Aula 3 – Trabalhando com Estrutura de Dados (Lista, Tuplas, Dicionários, Conjuntos, Trabalhando com Strings e Manipulação de Arquivos)