Educação Profissional Paulista

Técnico em
De sen volvim en to
de Sistem a s



Estruturas de decisão simples **Comandos condicionais** Aula 1 [SIS]ANO1C1B2S11A1



Objetivos da Aula:

Conhecer as funções dos comandos condicionais if e else no contexto da linguagem Python e estruturas de decisão, com exemplos demonstrativos.



Competências da Unidade (Técnicas e Socioemocionais):

Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento. Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes. Trabalhar a curiosidade e resiliência em resolução de problemas computacionais.



Recursos Didáticos:

- Recurso audiovisual para exibição de vídeos e imagens;
- Caderno, canetas, lápis.



Duração da Aula:

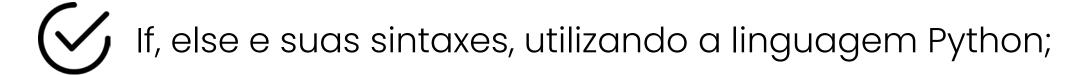
50 minutos



Desenvolvimento da aula

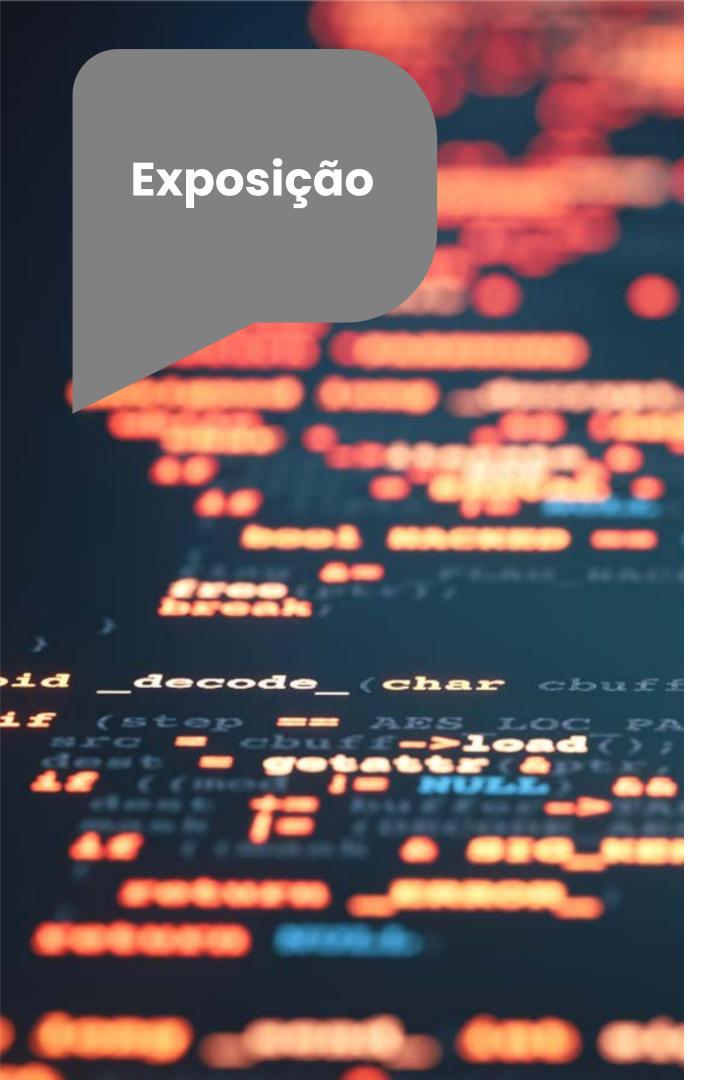
Nesta aula, abordaremos:











Introdução - Python

Comandos condicionais:

- são estruturas que permitem decisões baseadas em condições específicas;
- permitem a execução de diferentes ações em diferentes circunstâncias;
- controlam o fluxo do programa;
- tornam o programa mais flexível e poderoso.



Comando if

- É a **estrutura de decisão** mais básica em Python;
- Serve para verificar se uma condição é verdadeira. Se for, executa um bloco de código específico. Veja os exemplos:

Verificar se um número é positivo
num = int(input("Digite um número: "))
if num > 0: print("O número é positivo.")

Verificar se um número é par
num = int(input("Digite um número: "))
if num % 2 == 0: print("O número é par.")



Expressões em Python

 Expressões booleanas são verdadeiras/falsas em comandos condicionais;

Exemplo:

```
# Expressão booleana
x = 5 > 3 # True
```

- Expressões **condicionais** usam operadores de comparação para valores diferentes;
- Expressões **condicionais** retornam valores booleanos usados em comandos if.

Exemplo:

```
# Expressão condicional
y = 5 == 3 # False
```



Clareza dos comandos

A **indentação** define a **estrutura** do código e delimita blocos de código dentro de **condicionais**.

A clareza na escrita é importante para facilitar a leitura por outros programadores.



Atenção!

A **falta** de indentação ou clareza pode causar **erros** de sintaxe e **dificultar** a **manutenção** do código.



Comparativo

```
# Código com indentação correta
idade = int(input("Digite sua idade: "))
if idade >= 18:
    print("Você é maior de idade.")
else:
    print("Você é menor de idade.")
```

```
# Código com indentação incorreta
idade = int(input("Digite sua idade: "))
if idade >= 18:
print("Você é maior de idade.")
else:
print("Você é menor de idade.")
```



Exemplo de código

Você vai ao cinema assistir a um filme, mas a classificação indicativa é de 12 anos.

Ao analisar o seu documento, qual seria um código simples que permitiria ou não a venda do ingresso?

Exemplo:

```
idade = int(input("Digite sua idade: "))
classificacao_indicativa = 12

if idade >= classificacao_indicativa:
    print("Você pode assistir ao filme!")
else:
    print("Desculpe, você não pode assistir a esse filme.")
```





Inteligências artificiais e automações

Qual é a automação mais simples encontrada em casas inteligentes atualmente?

Nesse contexto, a tomada de decisão sempre trafega entre duas opções. Quais são?

Como você definiria os requisitos de um programa para a montagem de uma casa inteligente?





Vamos fazer uma **atividade**

Produção de texto síntese do debate

O texto deve ter entre 250 e 500 caracteres com espaços (equivalente a 5 ou 10 linhas)



15 minutos



Individual



Vale nota

- Mantenha o texto **centrado no assunto** principal do debate e inclua a conclusão ou ideia principal logo no início do texto.
- Resuma os argumentos ou pontos de vista mais significativos da discussão. Seja conciso e direto.
- Estruture seu texto de maneira clara e lógica.
 Comece com uma introdução ao tema,
 seguida de suas reflexões, e conclua com
 uma ideia final.
- Use **linguagem clara e compreensível**. Antes de entregar, revise para garantir correção gramatical, clareza e coerência das ideias.





Hoje desenvolvemos:

O conhecimento do papel das estruturas condicionais ifelse na tomada de decisões dentro do fluxo de programas;

2 O entendimento do início da organização de código a partir dos conceitos vistos em indentação;

3 A prática de exercícios e debate sobre tecnologias atuais e aplicação de lógica computacional nesse contexto.





Referências da aula

PINHO, M. S. *Programação em Python*: Comandos de Decisão/Comandos de Seleção. Escola Politécnica – PUCRS. Disponível em:

https://www.inf.pucrs.br/pinho/PCB/ComandosDeDecisao/Decisao.htm. Acesso em: 3 mar. 2024.

HASHTAG PROGRAMAÇÃO. *Como Utilizar as Operações Básicas no Python*. Disponível em: https://www.youtube.com/watch?v=Z2_xYFWvINg. Acesso em: 3 mar. 2024.

Identidade visual: Imagens © Getty Images



Educação Profissional Paulista

Técnico em
De sen volvim en to
de Sistem a s

