## Educação Profissional Paulista

Técnico em

Desenvolvimento

de Sistemas





#### Exposição



#### Objetivos da Aula

 Compreender quais são as principais alternativas ao uso do comando SWITCH.



#### Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver sistemas computacionais, utilizando ambiente de desenvolvimento.
- Trabalhar a criatividade e a resolução de problemas computacionais.



#### **Recursos Didáticos**

- Recurso audiovisual para exibição de vídeos e imagens.
- Folhas sulfite, canetas coloridas e lápis.



#### Duração da Aula

50 minutos

#### Exposição

#### Alternativas ao SWITCH e futuro da programação

Este tema explora alternativas ao SWITCH e discute tendências futuras em programação condicional, preparando os leitores para evoluções na área.



Alternativas ao comando SWITCH (exemplo: Pattern Matching).

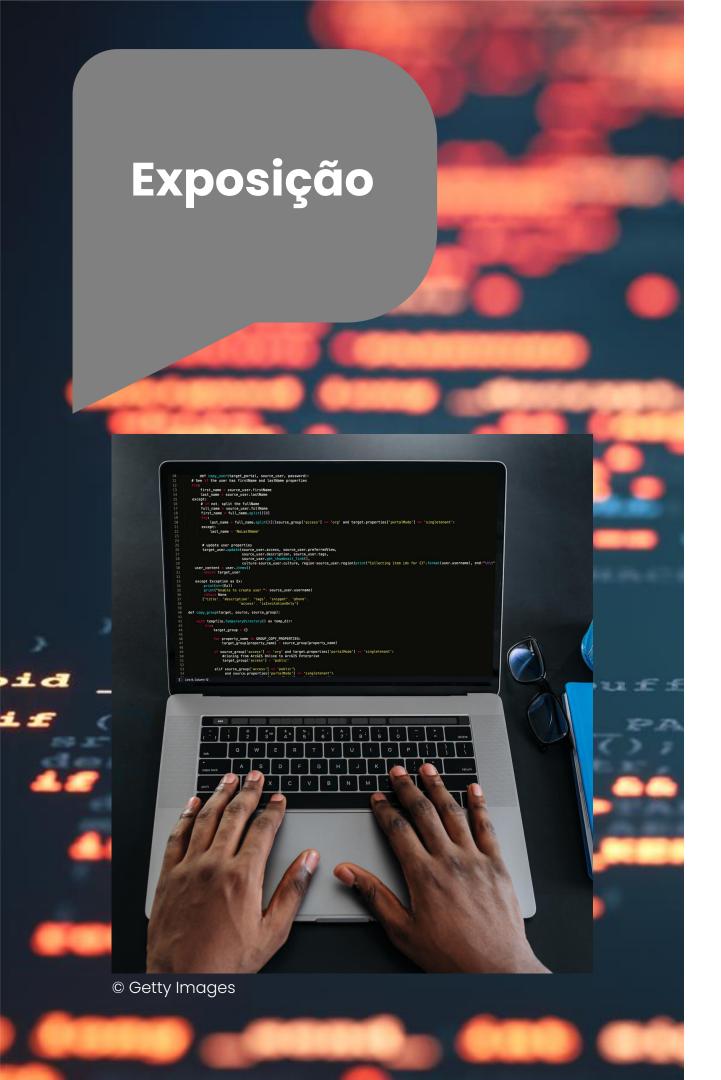


Vantagens e desvantagens dessas alternativas.



Tendências futuras em estruturas condicionais.





#### Alternativas ao comando SWITCH

O comando SWITCH é uma estrutura condicional comum em diversas linguagens de programação, como C, C++, e Java, mas Python não dispõe de um comando nativo SWITCH. Em Python, as alternativas comuns incluem o uso de:

**Instrução if-elif-else:** é a forma mais básica e direta de substituir um *SWITCH*. Funciona bem para um número limitado de casos.

**Dicionários**: permitem mapear chaves a funções ou valores, atuando como um SWITCH. São úteis para um grande número de casos e quando as ações são mais complexas.

Pattern Matching (introduzido no Python 3.10): comparação (ou correspondência) de padrões, do inglês Pattern Matching, é um dos conceitos fundamentais da programação funcional. Este conceito pode substituir uma série de desvios de fluxo de código, como:

- Comparações IF-ELSE;
- Switch-cases;
- Laços de repetição (for, for each, while e do);
- Comparação de tipos.



#### Exposição

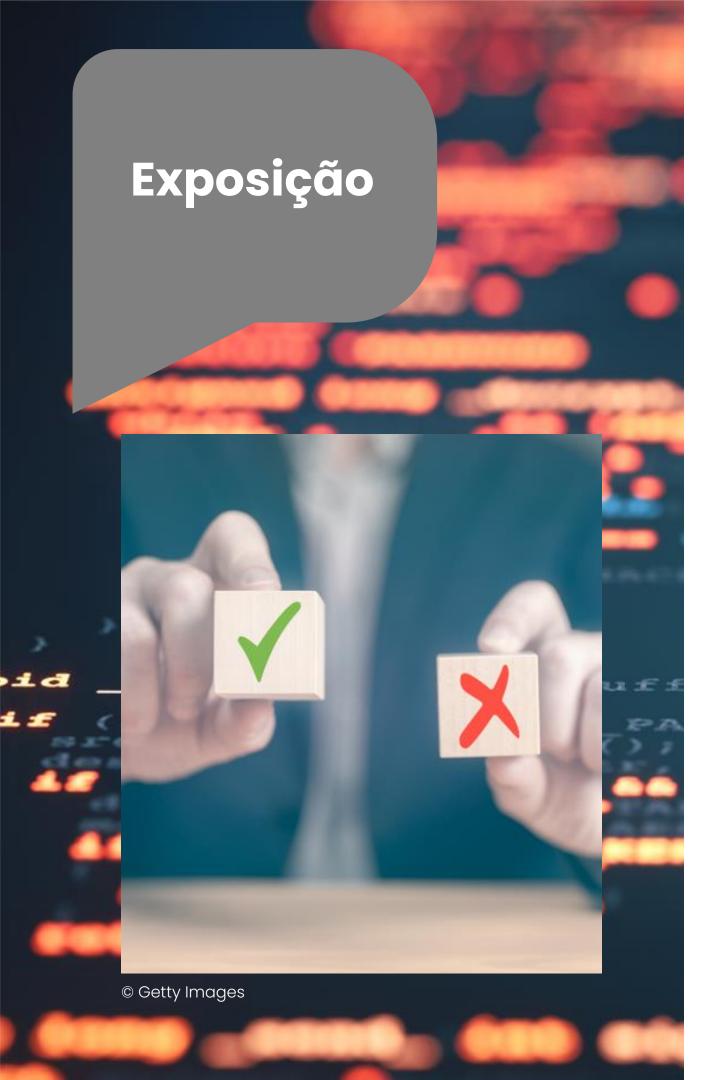
#### Exemplos

# If-elif-else: def exemplo\_if\_else(valor): if valor == 'A': return "Opção A selecionada" elif valor == 'B': return "Opção B selecionada" else: return "Outra opção" print(exemplo\_if\_else('A')) # Saída: Opção A selecionada

```
Dicionários:
def opcao_a():
  return "Opção A executada"
def opcao_b():
  return "Opção B executada"
switch_dict = {
  'A': opcao_a,
  'B': opcao_b
valor = 'A'
print(switch_dict.get(valor, lambda:
  "Opção inválida")())
# Saída: Opção A executada
```

# Exemplo com Pattern Matching: def exemplo\_pattern\_matching(valor): match valor: case 'A': return "Opção A" case 'B': return "Opção B" case \_: return "Outra opção" print(exemplo\_pattern\_matching('A')) # Saída: Opção A





#### Vantagens e desvantagens

#### If-elif-else

Vantagens: simples e direto, fácil de entender.

Desvantagens: pode se tornar verboso e difícil de manter com muitas condições.

#### Dicionários

Vantagens: rápido para um grande número de casos, flexível (permite associação de funções).

Desvantagens: menos intuitivo, requer inicialização do dicionário.

#### **Pattern Matching**

Vantagens: muito flexível, permite correspondência complexa de padrões.

Desvantagens: mais novo em Python, pode ser menos familiar para alguns desenvolvedores.





### Tendências futuras em estruturas condicionais

**Aumento do uso de** *Pattern Matching*: com o tempo, espera-se um uso mais amplo dessa abordagem, devido à sua flexibilidade e ao seu poder.

Integração com Inteligência Artificial: estruturas condicionais podem se tornar mais dinâmicas e adaptativas, utilizando IA para otimizar decisões.

#### Linguagens Específicas de Domínio (DSLs):

desenvolvimento de linguagens mais orientadas a casos de uso específicos, incluindo estruturas condicionais personalizadas.



#### Vamos fazer uma atividade

Você trabalha com desenvolvimento de software e acaba de receber uma mensagem em seu slack de Laura. Ela trabalha com você e está tendo o primeiro contato com estrutura de seleção.





Em grupos de até cinco pessoas

### Montagem do fluxo de pagamento

- A mensagem diz: Olá! Recebi a tarefa de criar um fluxo de seleção para os pagamento da loja por meio de algoritmos e *Python*, mas não tenho muita experiência com essa demanda. Por gentileza, você pode me ajudar?
- **O objetivo desta atividade** é que você forneça à Laura os passos principais de um fluxo de pagamento em *Python*, em uma loja de esportes.
- Para respondê-la, monte uma mensagem pensando no formato do slack (chat empresarial), explicando de forma sucinta e objetiva os passos que devem ser seguidos.

  Considere, por exemplo, que a pessoa pode optar por pagamento em cartão ou dinheiro, o que gera a cobrança por diferentes meios. Essas decisões são muito importantes.





#### Hoje desenvolvemos:

- A compreensão sobre **as principais tendências futuras em estruturas condicionais**;
- 2 O conhecimento sobre quais as vantagens e as desvantagens dessas alternativas;
- 3 A identificação das **alternativas ao comando SWITCH** (exemplo: Pattern Matching).



#### Referências da aula

LET'S DATA. *If, Elif, Else* (Estruturas de Decisão) | *Python* em 30 minutos. Disponível em: <a href="https://www.youtube.com/watch?v=Yvo1lHk3QmA&t=953s">https://www.youtube.com/watch?v=Yvo1lHk3QmA&t=953s</a>. Acesso em: 21 mar. 2024.

VALE, J. C. S. *Estruturas de Seleção em Python* – #07. DEV, 2022. Disponível em: <a href="https://dev.to/jcarlosvale/estruturas-de-selecao-em-python-07-2mac">https://dev.to/jcarlosvale/estruturas-de-selecao-em-python-07-2mac</a>. Acesso em: 21 mar. 2024.

Identidade visual: Imagens © Getty Images.

## Educação Profissional Paulista

Técnico em

Desenvolvimento

de Sistemas

