

Educação
Profissional
Paulista

Técnico em
Desenvolvimento
de Sistemas

Estruturas de Seleção

Introdução às Estruturas de Seleção

Aula 2

Código da aula: [SIS]ANO1C1B1S4A2

Exposição



Objetivos da aula

- Compreender o que são as estruturas de seleção;
- Conhecer a diferença entre as aplicações de estruturas de seleção na lógica de aplicações Python.



Competências da Unidade (técnicas e socioemocionais)

- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento;
- Identificar e analisar problemas, desenvolver alternativas e implementar soluções eficazes durante a execução de um projeto.



Recursos didáticos

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



Duração da aula

50 minutos.

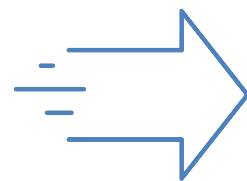
Desenvolvimento da aula

Antes de iniciar a aula, pergunte aos alunos quais foram os conteúdos trabalhados na última aula. Verifique se ainda restam dúvidas e se os alunos compreenderam os conceitos estudados.

- ✓ Compreender o conceito de estruturas de seleção na programação.
- ✓ Identificar situações do cotidiano em que as estruturas de seleção são úteis.
- ✓ Aprender a implementar estruturas de seleção em linguagens de programação.

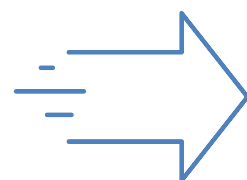
switch case e if-elif-else

As estruturas de seleção são fundamentais em programação, pois permitem que o fluxo do programa seja condicionalmente alterado com base em certas condições. As duas mais comuns em diferentes linguagens de programação são o ***switch case*** e o ***if-elif-else***.



Switch case

É uma estrutura de seleção encontrada em linguagens como C++, Java, C#, entre outras. Ela é especialmente útil quando se deseja testar o valor de uma única variável e realizar diferentes ações com base nesse valor.



If-elif-else

É outra estrutura de seleção comum em linguagens como Python, C++, Java, entre outras. Ela é mais flexível que o *switch case*, pois permite testar várias expressões e condições complexas.

Switch case e if-elif-else

A sintaxe geral das estruturas de seleção é a seguinte:

switch case no C++

```
switch (expressão) {  
    case valor1:  
        // código a ser executado quando  
        expressão == valor1  
        break;  
    case valor2:  
        // código a ser executado quando  
        expressão == valor2  
        break;  
    // casos adicionais  
    default:  
        // código a ser executado quando  
        nenhum dos casos anteriores é verdadeiro  
}
```

if-elif-else no Python

```
if expressão1:  
    # código a ser executado quando  
    expressão1 é verdadeira  
elif expressão2:  
    # código a ser executado quando  
    expressão2 é verdadeira  
# mais elifs, se necessário  
else:  
    # código a ser executado quando todas  
    as expressões anteriores são falsas
```

Switch case e if-elif-else

A sintaxe geral das estruturas de seleção é a seguinte:

switch case no C++

```
var opcaoUsuario; // valor obtido por alguma entrada do usuário

switch (opcaoUsuario) {
  case 1:
    console.log("Você escolheu receber e-mails de propaganda.");
    // código adicional para inscrever o usuário na lista de e-mails
    break;
  case 2:
    console.log("Você escolheu não receber e-mails de propaganda.");
    // código adicional para remover o usuário da lista de e-mails
    break;
  default:
    console.log("Opção inválida. Por favor, escolha uma opção válida.");
    // código adicional para lidar com entradas inválidas
}
```

if-elif-else no Python

```
int(input("Digite 1 para receber e-mails de propaganda ou 2 para não receber:"))
if opcao_usuario == 1:
    print("Você escolheu receber e-mails de propaganda.")
    # código adicional para inscrever o usuário na lista de e-mails
elif opcao_usuario == 2
    print("Você escolheu não receber e-mails de propaganda.")
    # código adicional para remover o usuário da lista de e-mails
else:
    print("Opção inválida. Por favor, escolha uma opção válida.")

# código adicional para lidar com entradas inválidas
```




© Getty Images

Exposição



Análise do vídeo

Com base no vídeo *Estrutura de decisão*, vamos fazer uma análise sobre estruturas de decisão e responder à questão nos slides seguintes:



SHARPAX. **Aula 08 – Estrutura de Decisão IF / ELSE | Lógica de Programação.**

YouTube, 12 jul. 2019. Disponível em:

<https://www.youtube.com/watch?v=juQgzKcbnuQ>. Acesso em: 16 jan. 2024.

Vamos
fazer um
quiz



Em um programa de computador, qual é o propósito de uma estrutura de decisão simples?

Executar um bloco de código
várias vezes até que uma
condição seja atendida.

Permitir ao programa
escolher entre duas ou mais
opções de execução.

Executar um bloco de código
somente se uma condição
específica for verdadeira.

Organizar o código em
funções para reutilização e
melhor manutenção.



Vamos
fazer um
quiz

**Em um programa de computador,
qual é o propósito de uma
estrutura de decisão simples?**

Executar um bloco de código
várias vezes até que uma
condição seja atendida.

✓
Permitir ao programa
escolher entre duas ou mais
opções de execução.

Executar um bloco de código
somente se uma condição
específica for verdadeira.

Organizar o código em
funções para reutilização e
melhor manutenção.

RESPOSTA CORRETA!

Em uma estrutura de decisão simples, um bloco de código é executado somente se uma condição específica for verdadeira, o que é o propósito fundamental.



Vamos
fazer um
quiz

Em um programa de computador, qual é o propósito de uma estrutura de decisão simples?



**Executar um bloco de código
várias vezes até que uma
condição seja atendida.**

RESPOSTA ERRADA! A descrição dada se refere a estruturas de repetição (como *loops*), e não a uma estrutura de decisão simples.



**Executar um bloco de código
somente se uma condição
específica for verdadeira.**

RESPOSTA ERRADA! A descrição se aplica a estruturas de decisão múltipla, como *"if-elif-else"*, onde há mais de uma opção de execução.



**Permitir ao programa escolher
entre duas ou mais opções de
execução.**

RESPOSTA CORRETA! Em uma estrutura de decisão simples, um bloco de código é executado somente se uma condição específica for verdadeira, o que é o propósito fundamental.



**Organizar o código em
funções para reutilização e
melhor manutenção.**

RESPOSTA ERRADA! A descrição se refere à modularização do código em funções, não à lógica de decisão simples.



© Getty Images

Exposição



Que tal aprendermos um pouco mais sobre estruturas de decisão com Python?



LET'S DATA. If, Elif, Else (Estruturas de Decisão) | Python em 30 minutos. YouTube, 6 jul. 2021. Disponível em: <https://www.youtube.com/watch?v=Yvo1lHk3QmA>. Acesso em: 16 jan. 2024.



O que nós
**aprendemos
hoje?**

Hoje desenvolvemos:

- 1** A compreensão de que o *switch case* é apropriado para avaliar o valor de uma única variável, enquanto o *"if-elif-else"* oferece maior flexibilidade ao permitir a avaliação de várias expressões complexas em diferentes cenários de programação;
- 2** O entendimento de que o *switch case* é uma escolha preferencial em situações onde há muitos casos diferentes a serem avaliados e quando os valores a serem testados são inteiros ou são caracteres, facilitando a estruturação do código de forma mais eficiente e legível;
- 3** A compreensão de que o *"if-elif-else"* é uma estrutura de seleção mais flexível e adequada para uma ampla variedade de situações, especialmente quando as condições de decisão são mais complexas, incluindo combinações lógicas de expressões. Isso permite uma abordagem mais versátil para o controle do fluxo do programa.

© Getty Images



Saiba mais

Vamos praticar um pouco mais!

CODE GAME. Disponível em:

<https://code.game/home>. Acesso em: 16 jan. 2024.

```
void _decode_(char cbuff **buff)
{
    if (step == AES_LOC_PASS) {
        src = cbuff->load();
        dest = getattr(&ptr, &mod,
            if (mod != NULL) as dest)
        dest += buffer->TABLE(mod)
        mask |= (previous_aes12
            if (mask & SIG_KERNEL) !=
        return _ERROR_
        return mask
    }
}
```


Referências da aula

REPOSITÓRIO PÚBLICO DO PROF. VICTOR MACHADO. Estruturas de Seleção, [s.d.]. Disponível em: https://victor0machado.github.io/prog/notas_aula/006-estruturas-seleção. Acesso em: 16 jan. 2024.

CURSO EM VÍDEO. Curso Python #014 – Estrutura de repetição while. YouTube, 21 nov. 2017. Disponível em: <https://www.youtube.com/watch?v=LH6OIn2IBaI>. Acesso em: 16 jan. 2024.

Identidade visual: Imagens © Getty Images.

Educação
Profissional
Paulista

Técnico em
Desenvolvimento
de Sistemas