Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Estrutura de seleção

Comandos condicionais: *SWITCH*

Aula 2

Código da aula: [SIS]ANO1C1B2S13A2

Exposição

Objetivos da Aula

- Conhecer implementações de utilização do SWITCH em cenários mais complexos de desenvolvimento de software.

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

SWITCH avançado e casos de uso

Este tema explora aplicações mais complexas do **SWITCH,** mostrando sua **flexibilidade e eficiência em diferentes cenários**.

- Uso avançado do SWITCH em cenários complexos.
- SWITCH em *loops* e com tipos de dados variados.
- **Estudo de caso:** aplicando SWITCH em um projeto real.

Vídeo

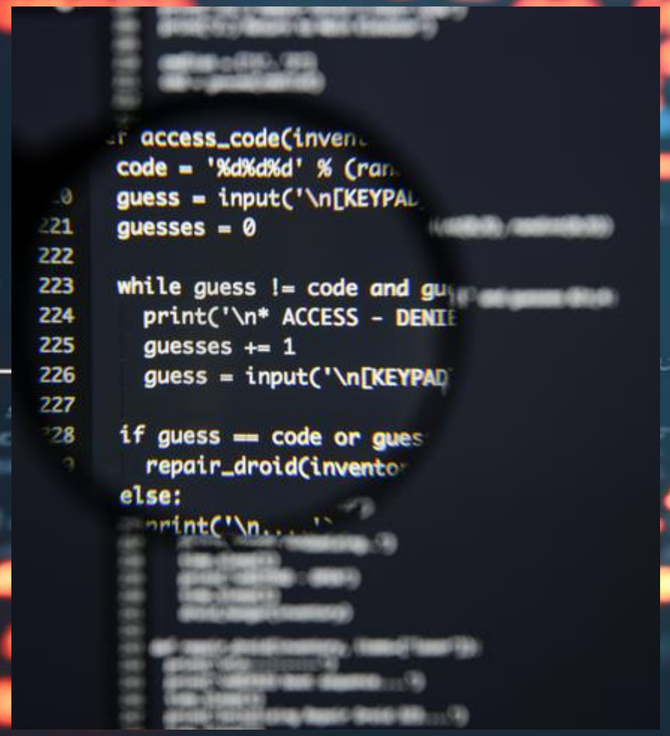
© Getty Images

Exposição

Que tal conhecer a plataforma Colab do Google?

ALURA. *Python para Data Science*: primeiros passos. Começando com Python. Utilizando o Google Colab. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122383. Acesso em: 20 mar. 2024.

Exposição



© Getty Images

Uso avançado do *SWITCH* em cenários complexos

Em Python, um padrão comum para simular um SWITCH é usar um dicionário no qual as chaves representam os casos, e os valores são as funções que executam o que seria o código de cada caso.

```
def processar_texto(texto):
    return texto.upper()

def processar_numero(numero):
    return numero * 2

def padrao():
    return "Opção inválida"

switch = {
    "texto": processar_texto,
    "numero": processar_numero
}
```

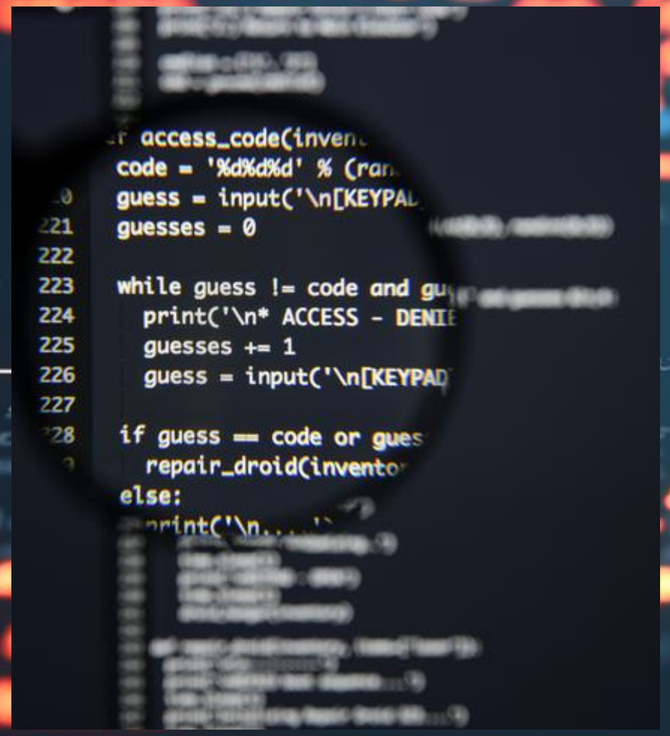
```
# Exemplo de uso
entrada = "texto“

valor = "Olá Mundo“

resultado = switch.get(entrada, padrao)(valor)

print(resultado)  # Saída: OLÁ MUNDO
```

Exposição



SWITCH em *loops* e com tipos de dados variados

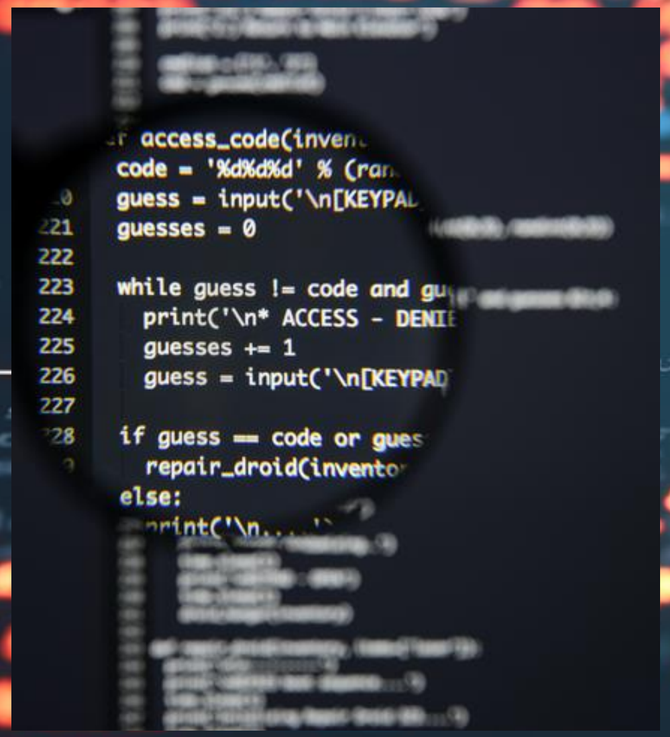
Podemos combinar SWITCH com *loops* para processar uma lista de tarefas ou dados de diferentes tipos. Vejamos um exemplo:

```
tarefas = [
    ("texto", "Olá"),
    ("numero", 10),
    ("desconhecido", None)
]

for tipo, valor in tarefas:
    resultado = switch.get(tipo, padrao)(valor)
    print(f"Resultado: {resultado}")
```

Neste código, processamos uma lista de tarefas, na qual cada tarefa tem um tipo e um valor. Usamos o SWITCH para determinar a função a ser chamada para cada tipo.

Exposição



Aplicando *SWITCH* em um projeto real

Imagine um sistema de processamento de comandos no qual cada comando deve ser executado de forma diferente.

Vamos criar uma simulação:

```
def adicionar_usuario(dados):
    # Lógica para adicionar um usuário
    return f"Usuário {dados['nome']} adicionado."

def remover_usuario(dados):
    # Lógica para remover um usuário
    return f"Usuário {dados['nome']} removido."

comandos = {
    "adicionar": adicionar_usuario,
    "remover": remover_usuario
}
```

 # Exemplo de uso em um sistema

```
comando_entrada = {"tipo": "adicionar", "dados": {"nome": "João"}}

comando = comando_entrada["tipo"]

dados = comando_entrada["dados"]

resultado = comandos.get(comando, padrao)(dados)

print(resultado)  # Saída: Usuário João adicionado.
```

Exposição

Aplicando *SWITCH* em um projeto real

Neste caso, **dependendo do tipo de comando recebido, uma função específica é chamada para lidar com esse comando, facilitando a extensão e a manutenção do código.**

Embora **Python não tenha um comando SWITCH nativo, a abordagem com dicionários e funções é uma maneira poderosa e flexível** de simular essa funcionalidade. **Esta técnica é particularmente útil em cenários nos quais diferentes tipos de processamento são necessários, com base em algum tipo de entrada**, como em sistemas de comandos ou processamento de dados variados.

Registro

Análise da aplicação do padrão *SWITCH* em *Python* em um cenário de negócios

Objetivo: desenvolver a capacidade analítica dos alunos ao examinar o uso do padrão SWITCH em Python em um contexto de negócios. Aperfeiçoar habilidades de escrita técnica e compreensão da aplicação prática de estruturas de controle de fluxo em programação.

Enunciado: Nesta atividade, **você fará a análise de um cenário hipotético de negócios no qual o padrão SWITCH, simulado por meio de dicionários e funções em Python**, poderia ser aplicado para melhorar a eficiência e a clareza do código. Você escreverá um **relatório, detalhando** como essa abordagem poderia ser implementada e os benefícios potenciais que ela traria.

Registro

Vamos fazer uma **atividade**

Análise de contexto em *Python* em um cenário de negócios. Não se esqueça de enviar o relatório no AVA.

25 minutos

Em grupos de até seis pessoas ou individualmente

Análise da aplicação do padrão *SWITCH* em *Python* em um cenário de negócios

1 **Analise o seguinte cenário hipotético:** uma empresa de e-commerce usa um sistema para processar diferentes tipos de solicitações de clientes (por exemplo, processar pedidos, retornar produtos, atualizar informações do cliente). **Imagine como o sistema atual poderia ser melhorado com a implementação de um SWITCH simulado em Python.**

2 Detalhe como você implementaria o padrão SWITCH neste cenário. Inclua exemplos de como os comandos seriam mapeados e processados.

Discuta as vantagens dessa abordagem em comparação com estruturas de controle tradicionais (if-elif-else).

3 Escreva um relatório abrangendo:

- Uma descrição do cenário e do sistema atual;
- Uma proposta detalhada de implementação do padrão SWITCH;
- Análise das vantagens e possíveis desafios desta abordagem;
- Reflexões sobre a aplicabilidade e a eficiência do padrão SWITCH em diferentes contextos de negócios;
- O relatório deve ser claro, conciso e bem estruturado.

Registro

Exemplo de resolução

Descrição do cenário atual: a empresa em questão dispõe de um sistema que lida com várias solicitações de clientes, como processamento de pedidos, devolução de produtos e atualização de informações do cliente. Atualmente, o sistema utiliza uma série de instruções if-elif-else para gerenciar essas tarefas, o que torna o código extenso, difícil de manter e propenso a erros.

Proposta de implementação do padrão *SWITCH*: para melhorar a eficiência e a clareza do código, propõe-se a implementação de um padrão SWITCH, usando dicionários e funções em Python. Este método simula o comportamento de um switch-case encontrado em outras linguagens de programação.

Vantagens da abordagem proposta: eficiência de código – reduz a complexidade do código, tornando-o mais legível e fácil de manter.

Escalabilidade: facilita a adição de novos casos sem alterar a estrutura lógica existente.

Manutenção: simplifica a depuração e a manutenção do código.

Desafios potenciais: curva de aprendizado – requer que os desenvolvedores estejam confortáveis com conceitos avançados em Python, como funções de primeira classe e dicionários.

Performance: em cenários com um número muito grande de casos, pode haver preocupações com a eficiência da busca no dicionário.

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. A compreensão sobre o **uso avançado do SWITCH em cenários complexos;**

2. O conhecimento sobre **a utilização do SWITCH em *loops* e com tipos de dados variados;**

3. Na prática, um estudo de caso: **aplicando SWITCH em um projeto real.**

Saiba mais

Agora, **vamos aprofundar um pouco mais em Python para interpretar as estruturas de seleção e suas principais aplicações:**

CURSOS KANE CHAN. *Estruturas de seleção If e Else em Python*. Disponível em: https://www.youtube.com/watch?v=zouf7AkISR4. Acesso em: 21 mar. 2024.

Referências da aula

ALURA. *Python para Data Science*: primeiros passos. Começando com *Python*. Utilizando o Google Colab. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122383. Acesso em: 20 mar. 2024.

CURSOS KANE CHAN. *Estruturas de seleção If e Else em Python*. Disponível em: https://www.youtube.com/watch?v=zouf7AkISR4. Acesso em: 21 mar. 2024.

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

Identidade visual: Imagens © Getty Images.

Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO