

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Estruturas de Repetição

Atividades práticas – Criação de programas

Aula 3

Código da aula: [SIS]ANO1C1B2S10A3

Exposição



Objetivos da aula

- Conhecer maneiras performáticas de otimizar a utilização da estrutura de repetição *while*.



Competências da unidade (técnicas e socioemocionais)

- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento;
- Resolver problemas computacionais com estratégias criativas.



Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Caderno para anotações.



Duração da aula

50 minutos.

Eficiência e otimização do laço *while*

A otimização do laço *while* é essencial para melhorar o desempenho dos programas e economizar recursos, especialmente em loops que processam grandes volumes de dados.

- ✓ Praticar aninhamento e escopo de variáveis por meio de algoritmos;
- ✓ Conhecer técnicas para minimizar o tempo de execução e o uso de memória;
- ✓ Técnicas de depuração para laços *while*.

Exposição



© Getty Images

Vamos lembrar o que é aninhamento e escopo de variáveis?

O **aninhamento** ocorre quando uma estrutura de controle é colocada dentro de outra, como um laço *while* dentro de outro laço *while* ou dentro de uma instrução *if*. Isso é útil quando se lida com situações que exigem várias camadas de decisão ou repetição.

Escopo de uma **variável** refere-se à parte do programa onde a variável é acessível. Variáveis definidas dentro de um laço têm escopo local; elas só podem ser acessadas dentro desse laço. Por outro lado, variáveis definidas fora de todos os laços têm escopo global, podendo ser acessadas de qualquer lugar do programa.

Exposição

Aninhamento e escopo de variáveis



© Getty Images

Exemplo de escopo em Python:

```
x = "global"
```

```
# Loop for para iterar duas vezes
```

```
for i in range(2):
```

```
    if i == 0:
```

```
        y = "local na primeira iteração"
```

```
    else:
```

```
        y = "local na segunda iteração"
```

```
print(f"Iteração {i+1}:")
```

```
print(f"  x: {x}") # Pode acessar x
```

```
print(f"  y: {y}") # Pode acessar y, pois está definida dentro do loop
```


Exposição



© Getty Images

Técnicas de depuração para refatoração

Depurar laços *while* pode ser desafiador, especialmente se eles não estão funcionando como o esperado. Aqui estão algumas técnicas:

Use um depurador: ferramentas de depuração permitem que você pause a execução do programa, inspecione variáveis e avance passo a passo através do código.

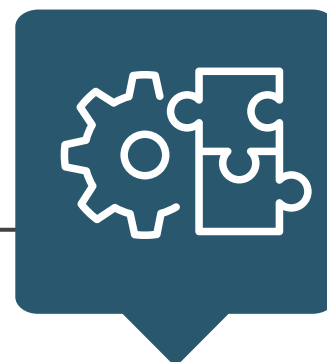
Refatoração: se você encontrar um laço particularmente complexo ou propenso a erros, pode ser um sinal de que o laço precisa ser refatorado. Talvez você possa dividir o laço em funções menores ou reescrever a lógica de uma forma que faça mais sentido.

Vamos
fazer uma
atividade

Vamos trabalhar dois
exercícios práticos
sobre repetição.

 **35 minutos**

Exercícios práticos – Estrutura de repetição



Materiais necessários

Os materiais necessários para a
realização da atividade são:

- Computador com acesso à internet;
- Caderno ou folhas de sulfite para anotações e rascunhos;
- Lápis ou canetas para anotações.



Passo a passo

Para executar a atividade prática, siga
estes passos:

1. Abra o arquivo complementar com a explicação da atividade;
2. Leia com atenção o enunciado de cada exercício, anotando os pontos principais;
3. Crie e execute os códigos propostos nas ferramentas de desenvolvimento em Python.



Faça download da explicação da atividade

Hoje desenvolvemos

- 1 A prática do funcionamento da estrutura de aninhamento e escopo de variáveis.
- 2 A revisão das técnicas de depuração para laços *while* e seu funcionamento.
- 3 A refatoração de um código em Python, buscando oportunidades de melhoria contínua e velocidade de execução.

O que nós
**aprendemos
hoje?**

© Getty Images



Saiba mais

Para fixar o que você aprendeu em Python, assista ao vídeo a seguir e pratique com os exercícios interativos.

LUIS FELIPE ZENI. *Programação: resolução de exercícios de for/while em Python HD*. Disponível em:

<https://www.youtube.com/watch?v=3jE56j9BReo>.

Acesso em: 25 fev. 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 NULL
    }
}
```

Referências da aula

CURSOS KANE CHAN. *Estrutura de repetição while em Python*. Disponível em: <https://www.youtube.com/watch?v=iFYWrDMfVNo>. Acesso em 25 fev. 2024.

LUIS FELIPE ZENI. *Programação: resolução de exercícios de for/while em Python HD*. Disponível em: <https://www.youtube.com/watch?v=3jE56j9BReo>. Acesso em: 25 fev. 2024.

PYTHON ACADEMY. *Loops e estruturas de repetição no Python*. Disponível em: <https://pythonacademy.com.br/blog/estruturas-de-repeticao>. Acesso em: 25 fev. 2024.

Identidade visual: © Getty Images.

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**