

# PYTHON PARA QAs



## MÓDULO 2 - ESTRUTURAS DE CONTROLE E COLEÇÕES

1. Estruturas de Decisão (Simples, Aninhadas e Compostas);
2. Estruturas de Repetição (Loops: for, while);
3. Coleções:
  - Listas;
  - Tuplas;
  - Conjuntos;
  - Dicionários.
4. Compreensão de listas e iterações avançadas.

## AULA 4 - ITERAÇÕES AVANÇADAS E COMPREENSÃO DE LISTAS

### Objetivos da aula:

- Revisar os conceitos de laços de repetição e coleções
- Aprofundar o uso de loops (for, while)
- Explorar funções embutidas para otimizar iterações:
  - enumerate()
  - zip()
  - map()
  - filter()
- Aprender a compreensão de listas para reduzir código repetitivo e melhorar a performance.
- Trabalhar com listas aninhadas e iteração sobre dicionários.
- Aplicar esses conceitos a cenários reais utilizando csv.

## REVISÃO AULA 3

- **Estruturas de Controle:**
  - Condicionais (if, elif, else).
  - Loops (for, while).
- **Coleções em Python:**
  - Listas, Tuplas, Conjuntos e Dicionários.
  - Compreensão de Listas para otimização de código.
- **Atividade de Revisão**
  - Criar um programa de um gerenciador de execução de testes automatizados, onde o usuário pode visualizar os casos de teste, rodá-los e ver o resultado da execução.
  - Regras:
    - O sistema começa com uma lista predefinida de casos de teste.
    - O usuário pode selecionar um caso de teste para execução.
    - O programa simula um resultado (Pass ou Fail).
    - O resultado de cada teste executado é armazenado em um dicionário.
    - Ao final, o usuário pode visualizar o relatório com todos os testes executados.

## ITERAÇÃO AVANÇADA EM ESTRUTURAS DE DADOS

- Como percorrer listas e dicionários de forma eficiente.
- Diferença entre percorrer com índices e percorrer diretamente os valores.
- Uso de funções `enumerate()`, `zip()`, `map()`, `filter()`

## COMPARAÇÃO RESUMIDA

Função	O que faz?	Exemplo de Uso
<code>enumerate()</code>	Percorre uma lista obtendo o índice e o valor	Enumerar casos de teste
<code>zip()</code>	Percorre múltiplas listas ao mesmo tempo	Comparar listas de entradas e saídas esperadas
<code>map()</code>	Aplica uma função a todos os elementos de uma lista	Ajustar notas de testes ou converter valores
<code>filter()</code>	Filtra elementos de uma lista com base em uma condição	Filtrar apenas testes aprovados

## COMPREENSÃO DE LISTAS E ESTRUTURAS ANINHADAS

- Como criar listas otimizadas com menos código e melhor desempenho.
- Aplicações Práticas com csv

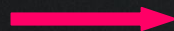
## REVISÃO

- Percorrer listas e dicionários de forma eficiente.
  - Diferença entre percorrer com índices e percorrer diretamente os valores.
  - Uso de funções `enumerate()`, `zip()`, `map()`, `filter()`
- Listas otimizadas com menos código e melhor desempenho.
- Aplicações Práticas com csv



# Podemos contar com o seu feedback?

Escaneie o QR Code ao lado e responda nossa Pesquisa de Avaliação.



# OBRIGADO



/apsferreira



/apsferreira\_



alura + FIA/P

PARA EMPRESAS

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.