

# PYTHON PARA QAs



## MÓDULO 3 - FUNÇÕES E MODULARIZAÇÃO

1. **Funções nativas e personalizadas em Python**
2. **Modularização de código e uso de bibliotecas**
  - **Pandas**
  - **NumPy**
3. **Estruturação de projetos para maior reutilização e organização.**

## AULA 6 - FUNÇÕES E MODULARIZAÇÃO

### Objetivos da aula:

- Entender boas práticas para organização de projetos Python
- Criar uma API REST simples utilizando Flask
- Testar essa API manualmente com requests e com pytest
- Implementar logging para capturar logs de requisições da API
- Utilizar Pandas e NumPy para processar respostas da API e gerar estatísticas

## REVISÃO AULA 5

- Funções personalizadas
- Funções embutidas.
- Escopo de variáveis
  - Variáveis Locais
  - Variáveis Globais
- Funções lambda
- Código em módulos reutilizáveis
- Bibliotecas externas (Request)

## REVISÃO AULA 5 - ATIVIDADES

- Criar um programa que:
  - Defina uma função para validar CPF.
  - Utilize um módulo externo (re para regex)
    - Regra: `re.match(r'^\d{3}\.\d{3}\.\d{3}-\d{2}$', cpf)`
  - Retorne se o CPF tem um formato válido ou inválido.
- Criar uma CLI de uma calculadora que utiliza um módulo Python que contenha funções matemáticas como:
  - Multiplicar
  - Somar
  - Diminuir
  - Dividir

## ESTRUTURAÇÃO DE PROJETOS PYTHON

- **O que um projeto bem estruturado precisa ter?**
  - Código modular (organizado em pacotes)
  - Testes automatizados
  - Documentação
  - Logging
- Estrutura de um Projeto Python para Testes Automatizados

- ```
/projeto_qa
  /src
    __init__.py
    api.py
    utils.py
  /tests
    test_api.py
  /logs
    api.log
  requirements.txt
  README.md
```

## CRIANDO UMA API REST COM FLASK

- **Por que criar uma API?**
  - APIs permitem que sistemas se comuniquem de forma programática.
  - Muitas ferramentas de QA dependem de APIs para validação de dados, testes de carga e integração contínua.
- **Instalando Flask**
  - `pip install flask`
- **Atividade:**
  - Criar um novo endpoint para apresentar o nome do curso.

## TESTANDO A API COM REQUESTS E PYTEST

- **Criando um teste com Requests para consumo manual da API**
- **Introdução ao pytest**
  - **O que é?**
    - O pytest é um framework para testes automatizados em Python, amplamente usado para testar funções, módulos e APIs.
  - **Por que usar?**
    - Simples e intuitivo
    - Automático
    - Flexível
    - Relatórios claros



## TESTANDO A API COM REQUESTS E PYTEST

- **Instalação e configuração do pytest.**
- **Criando testes unitários e de integração para APIs.**
- **Executando os testes e interpretando os resultados.**
- **Atividade prática detalhada para fixação.**

## REVISÃO

- Estruturamos um projeto Python organizado.
- Criamos e testamos uma API com Flask.
- Utilizamos requests para consumo manual de APIs.
- Automatizamos testes de API com pytest.
- Implementamos logging para rastrear chamadas à API.
- Manipulamos e analisamos respostas da API com Pandas e NumPy.

## ATIVIDADE

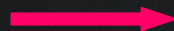
- Criar um Sistema de Consulta de Produtos com API e Testes Automatizados
  - Criar uma API REST com Flask
  - Manipular listas e dicionários
  - Testes automatizados com pytest
  - Uso da biblioteca requests para consumir APIs

## ATIVIDADE

- funcionalidades do Sistema:
  - Cadastrar um novo produto
    - Método: POST /produtos
    - id (int)
    - nome (str)
    - preco (float)
    - Retorno esperado: {"mensagem": "Produto cadastrado"}
  - Listar todos os produtos
    - Método: GET /produtos
    - Retorno esperado: Uma lista com os produtos cadastrados
  - Buscar um produto pelo ID
    - Método: GET /produtos/<id>
    - Retorno esperado: Dados do produto ou mensagem de erro caso não exista
  - Atualizar o preço de um produto
    - Método: PUT /produtos/<id>
    - Parâmetro: preco (float)
    - Retorno esperado: {"mensagem": "Preço atualizado"}
  - Remover um produto pelo ID
    - Método: DELETE /produtos/<id>
    - Retorno esperado: {"mensagem": "Produto removido"}

# 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.