

1

O que estudaremos?

Veremos:
Ferramentas de suporte ao desenvolvedor
Testes unitários (*unit test*)
Geração de executável



3

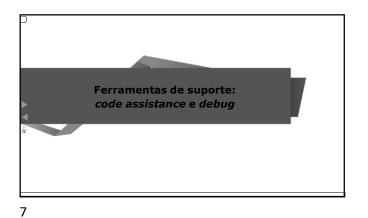
Code complete

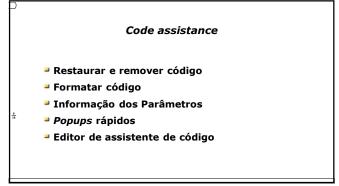
Básico
TAB
Pós-fixo
Type-mathing
F-strings

Code refactoring

Refatorar menu
Renomear
Extrair variáveis
Refatorar concertos rápidos
Refatorar no lugar

5 6

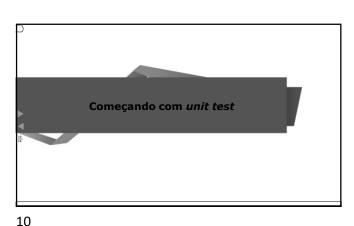




8

Debug

■ Configurando e entendo o fluxo do debug



9

O que é um teste unitário (unit test)

É uma biblioteca de teste unitário incorporada ao Python que fornece uma estrutura para escrever e executar testes automatizados

Os testes unitários são usados para verificar se partes específicas do código estão funcionando corretamente

Quando usar o unit test

Testar funções e métodos
Garantir a consistência
Facilitar a manutenção
Dar suporte à prática de desenvolvimento orientado a testes (TDD)
Partes críticas do código que são propensas a erros ou com dependências externas

11 12

## Quando usar o unit test Quando se trata de quando usar o unit test, é recomendado escrever testes unitários para partes críticas e complexas do código, especialmente aquelas que são propensas a erros ou que possuem dependências externas

PyTest

13 14

## PyTest É uma biblioteca baseada no unit test, mas com uma sintaxe mais simples e mais recursos: Sintaxe simplificada Asserções avançadas Descoberta automática de testes

PyTest

O pytest é uma biblioteca de teste que se baseia no *unit test*, mas oferece uma sintaxe mais simples e recursos avançados

15 16



Gerando executável

Para transformar um projeto Python em um executável no Windows, você pode usar ferramentas como PyInstaller ou cx\_Freeze

Ambas empacotam o seu código Python e geram um executável independente que pode ser executado sem a necessidade de ter o Python instalado

17 18

