

Resumo Completo - Fundamentos de Python + Pytest

RESUMO COMPLETO - FUNDAMENTOS DE PYTHON

1. VARIÁVEIS

Variáveis armazenam valores. Ex:

```
nome = "João"
```

```
idade = 30
```

2. TIPOS DE DADOS

- int: números inteiros

- float: números decimais

- str: textos

- bool: True ou False

3. STRINGS

```
mensagem = "Olá"
```

```
mensagem.upper(), mensagem.lower(), mensagem.split(), len(mensagem)
```

4. NÚMEROS E OPERAÇÕES

```
+, -, *, /, //, %, **
```

5. LISTAS

```
frutas = ["maçã", "banana"]
```

```
frutas.append("laranja"), frutas[0], len(frutas)
```

6. DICIONÁRIOS

```
pessoa = {"nome": "Ana", "idade": 25}
```

```
pessoa["nome"], pessoa["altura"] = 1.70
```

7. CONDICIONAIS

```
if idade >= 18:
```

```
print("Maior")
```

8. LAÇOS

```
for i in range(5), while condicao:
```

9. FUNÇÕES

```
def saudacao(nome):  
    return f"Olá, {nome}"
```

10. CLASSES

```
class Pessoa:  
    def __init__(self, nome):  
        self.nome = nome  
  
    def apresentar(self):  
        print(self.nome)
```

```
p = Pessoa("João")
```

```
=====
```

RESUMO COMPLETO - PYTEST

1. O QUE É PYTEST?

Framework de testes automatizados em Python.

2. TESTE BÁSICO

```
def soma(a, b): return a + b  
  
def test_soma(): assert soma(2, 3) == 5
```

3. ASSERTIONS

```
assert 1 == 1, assert "a" in "abc", assert [1, 2] != [2, 1]
```

4. FIXTURES

```
@pytest.fixture

def dados():

    return {"nome": "Maria"}

def test_nome(dados):

    assert dados["nome"] == "Maria"
```

5. ORGANIZAÇÃO

- Arquivos: test_*.py
- Pastas: use "tests/"

6. SETUP E TEARDOWN

```
def setup_function(): print("Setup")

def teardown_function(): print("Teardown")
```

7. PARAMETRIZAÇÃO

```
@pytest.mark.parametrize("entrada, esperado", [(2, 4), (3, 9)])

def test_quad(entrada, esperado):

    assert entrada ** 2 == esperado
```

8. MOCKS

```
from unittest.mock import Mock

api = Mock(); api.get.return_value = {"status": 200}
```

9. EXECUTANDO TESTES

```
pytest, pytest -v, pytest --html=relatorio.html
```

10. BOAS PRÁTICAS

- Nomeie bem os testes
- Use fixtures
- Testes pequenos e independentes