

# Anotação de Tipos

Prof. Alberto Costa Neto  
Programação em Python

# Erros de Tipos

- Em algumas situações, por uma falha do programador, ocorrem erros de tipos durante a execução de um programa Python. Por exemplo:

```
>>> nome = input('Digite o seu nome:')
Jose
>>> idade = int(input('Digite sua idade:'))
34
>>> print(nome + ' está com ' + idade + ' anos!')
Traceback (most recent call last):
  File "<python-input-2>", line 1, in <module>
    print(nome + ' está com ' + idade + ' anos!')
          ~~~~~^~~~~~
TypeError: can only concatenate str (not "int") to str
```

# Erros de Tipos

- A concatenação de uma variável string (**nome**) com uma variável int (**idade**) **não é aceita em Python**
- Como os tipos das variáveis são dinâmicos em Python, **só é possível detectar este erro ao executar o programa**



# LP's

## Estaticamente Tipadas

- Outras linguagens de programação possuem **definição de tipos de dados de variáveis**, tais como C, C++ e Java

```
int x = 0;  
String s = "ok";
```

- A checagem para verificar se não há erros referentes aos tipos é **feita em tempo de compilação** (antes de executar)

# Anotações de Tipos

- Anotações de tipos **surgiram na versão 3.5 de Python**.
- Oferecem uma maneira de **informar o tipo associado a um identificador**.
- Com base nos tipos anotados, é possível **inferir estaticamente se as regras de tipos foram respeitadas**
- Pode ser utilizado por **checadores de tipos**, IDEs e outras ferramentas, como **MyPy**, **Pyright** (Microsoft), **Pyre** (Meta), **Pytype** (Google) e outras embutidas nos IDEs.

# Anotando Tipos

 anotacao-tipos.py x

```
1 x: int = 10
2 y: float = 10.5
3 nome: str = 'Jose'
```

- Definindo **x** como **int**
- Definindo **y** como **float**
- Definindo **nome** como **str**

# Checagem de Tipos

anotacao-tipos.py x

```
1 x: int = 10
2 y: float = 10.5
3 nome: str = 'Jose'
4 print(x, type(x))
5 print(y, type(y))
6 print(nome, type(nome))
7
8 x = '10'
9 print(x, type(x))
10
11 concat_1: str = nome + ' ' + x
12 print(concat_1)
13
14 concat_errada: str = nome + y
15 print(concat_errada)
```

- Mensagem gerada pela checagem estática do **PyCharm 2025.2 CE**



- ⚠ Expected type 'int', got 'str' instead :8
- ⚠ Expected type 'str', got 'LiteralString | str | int' instead :11
- ⚠ Unexpected type(s):(int)Possible type(s):(LiteralString)(str) :11
- ⚠ Expected type 'str', got 'LiteralString | str | float' instead :14
- ⚠ Unexpected type(s):(float)Possible type(s):(LiteralString)(str) :14

# A importância de Anotação de Tipos

- Documentação das variáveis
- Disponibilidade de informações dos tipos para IDEs
- Detecção antecipada de erros de tipo
- Prevenção de alguns erros, como uma troca de parâmetros em chamadas de funções