# PYTHON, TIPOS & SEGURANÇA

Como *type hinting* e análise estática podem ajudar você e sua equipe

# OI, EU SOU O (FELIPE) BIDU!

- github.com/fbidu
- Desenvolvedor Back End
- Arquiteto de Infra no Poppin

# O QUE É UM TIPO?

# 01100001

- Qual o significado dessa cadeia binária?
- Em decimal, é o número 97
- Na tabela ASCII, é o caractere "a"

# OPERAÇÕES

- De acordo com *como* entendemos uma informação, as operações possíveis mudam!

```
>>> 97 * 10
970


>>> 97 - 55
42
```

# 01100001 == 97?

```
>>> "a" * 10
'aaaaaaaaaa'

>>> "a".upper()
'A'

>>> "a".isalpha()
True

>>> "a".isdigit()
False
```

# 01100001 == "a"?

Tipos permitem definir quais operações podemos ou não aplicar sobre um conjunto de informações!

```
>>> 80218 + "Python"

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

# Python:

```
>>> "Python Brasil " * 2
'Python Brasil Python Brasil '
```

# Java:

```
String a = "Python Brasil ";
a * 2;
|  Error:
|  bad operand types for binary operator '*'
|    first type:  java.lang.String
|    second type: int
|  a * 2
|  ^---^
```

*"[...] A type is a set of values and a set of functions that one can apply to these values."*

— Guido van Rossum, Ivan Levkivskyi - PEP 483

# SISTEMAS DE TIPOS

- Diversos nomes diferentes
- Estático vs. Dinâmico
- Forte vs. Fraco
- Explícito vs. Implícito
- (autor A) vs. (autor B)

# ESTÁTICO VS. DINÂMICO

O tipo de uma variável pode mudar?

# FORTE VS. FRACO

Quão "chata" a linguagem é com tipos?

# EXPLÍCITO VS.

# IMPLÍCITO

Você consegue saber o tipo de uma variável só lendo o código?

Python tem um sistema de tipos **implícito**:

```
>>> ano = 2019
```

Python tem um sistema de tipos implícito, **dinâmico**:

```
>>> ano = 2019
>>> ano = str(ano)
```

Python tem um sistema de tipos implícito, dinâmico e **forte**:

```
>>> ano = 2019
>>> ano = str(ano)
>>> ano / 10

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

# ANÁLISE ESTÁTICA & TIPAGEM DINÂMICA

Juntando dois pedaços da conversa de hoje

# PODEMOS CHECAR CÓDIGO DE

# DUAS FORMAS

1. Lendo o código
2. Rodando o código

| Método de Análise | Tipo |
| --- | --- |
| Leitura | Estática |
| Execução | Dinâmica |

# ANÁLISE ESTÁTICA

1. Livre de efeitos do código
2. Mais rápida de definir e executar
3. A cobertura é sempre 100%

# ⚠️⚠️ ALERTA ⚠️⚠️

Análise Dinâmica é MUITO importante, ela só não é o foco de hoje!

Análise Estática é muito boa, muito legal, mas...

...ela só é tão boa quanto tanto a linguagem consegue contar pra ela

# Tome esse pedaço de código em Java, por exemplo

```
int some_function(int a, int b) {
    return a.toUpperCase();
}
```

# Porém em Python...

```
def some_function(a, b):
    return a.upper()
```

*ALERTA DE FLAME WAR*

# TIPAGEM DINÂMICA É UM PROBLEMA?

Questão de design intrínseco à linguagem!

# JUNTANDOS OS DOIS MUNDOS

Como casar o dinamismo de um lado com a riqueza de informações do outro?

# GRADUAL TYPING!

Dinâmico quando desejado, estático quando necessário

Jeremy Siek, 2006

Essa é uma das referências mais importantes dessa palestra!

# NO MUNDO PYTHON...

- Debates sobre isso acontecem desde o ano 2000
- A primeira implementação que se aproxima disso é de 2006, o PEP 3107
- Referência legal: GvR - Type Hints - PyCon 2015

# FERRAMENTAS

# SOZINHAS NÃO FAZEM NADA!

Precisamos de **métodos**