



Informática Industrial

Introdução ao Python: Variáveis e Objetos

Prof. Guilherme Márcio Soares, Dr. Eng.
guilherme.marcio@ufjf.edu.br

Variáveis

❑ Em Python, todos os dados são objetos, inclusive os mais simples como inteiros, booleanos, etc.

❑ *Objetos mutáveis:*

- Objetos cujos valores **podem ser modificados**. Ex.: listas, dicionários, objetos definidos pelo usuário.

❑ *Objetos imutáveis:*

- Objetos cujos valores **não podem ser modificados**. Ex.: Números, tuplas, booleanos, strings.

**Impacta a
forma de
atribuição!!**



Informática Industrial

Introdução ao Python: Criação de sub-rotinas e módulos

Prof. Guilherme Márcio Soares, Dr. Eng.
guilherme.marcio@ufjf.edu.br

Criação de sub-rotinas

- ❑ Para se criar funções e sub-rotinas em Python utiliza-se a palavra-chave `def`.

```
def(arg1, arg2,argN):  
    """  
    docstring (opcional) -> documentação  
    """  
    expressões  
    return valor (opcional)
```

**Cuidado com
a
indentação !**

Criação de sub-rotinas

- ❑ Em Python utiliza-se o conceito de passagem por **atribuição** ou por **referência de objeto**.
- ❑ De forma simplificada pode-se resumir que:
 - Quando o **objeto passado como** argumento **é mutável**: *uma mudança da variável dentro da função provocará uma mudança na variável do método que a invocou;*

**Cuidado com
a
indentação !**

Criação de sub-rotinas

- ❑ Em Python utiliza-se o conceito de passagem por **atribuição** ou por **referência de objeto**.
- ❑ De forma simplificada pode-se resumir que:
 - Quando o **objeto passado como** argumento **é imutável**: *uma mudança da variável dentro da função **não** provocará uma mudança na variável do método que a invocou;*

**Cuidado com
a
indentação !**

Criação de módulos

❑ Módulos são arquivos que possuem um conjunto de **declarações** e **definições**.

❑ O nome do módulo é o nome do arquivo sem o “.py”.

❑ Exemplo:

- Nome do arquivo: **minhalib.py**
- Nome do módulo: **minhalib**

Criação de módulos

❑ Para se utilizar um módulo em um outro arquivo, basta usar a palavra **import**.

- Importar todo o módulo

```
import modulo
```

- Importar somente func1 e var1

```
from modulo import func1, var1
```

- Importar **var1** com o nome **variável**

```
from modulo import var1 as variavel
```


Criação de módulos

- ❑ Para executar um módulo como script, deve-se incluir o seguinte bloco de código:

```
if __name__ == "__main__":  
    expressões;
```

- ❑ Tudo que estiver declarado dentro do bloco **if** não será executado no **import**, mas somente quando o módulo for executado como script principal.



Informática Industrial

Introdução ao Python: POO – Classes e Herança

Prof. Guilherme Márcio Soares, Dr. Eng.
guilherme.marcio@ufjf.edu.br

POO: Classes

- ❑ O conceito de classes em Python é o mesmo do que em outras linguagens, ou seja, **classes são modelos para a criação de objetos**.
- ❑ Sintaxe:

```
class NomedaClasse:  
    def __init__(self,arg1,arg2):  
        self.attr1 = arg1  
        self.attr2 = arg2  
    def metodo1(self):  
        expressões  
    def metodo2(self,arg1,arg2):  
        expressões
```

POO: Classes

❑ Especificadores de acesso

- “__” (2x underscore) para atributos e métodos **privados**.
- “_” (1x underscore) para atributos e métodos **protegidos**.

❑ Acesso aos atributos e métodos do objeto

- O acesso aos atributos e métodos dentro das funções-membro pode ser feito pela palavra chave **self** (análoga ao ponteiro **this** do C++).

POO: Classes

❑ Construtor

- Método especial chamado **`__init__()`**
- É invocado automaticamente quando o objeto é instanciado.

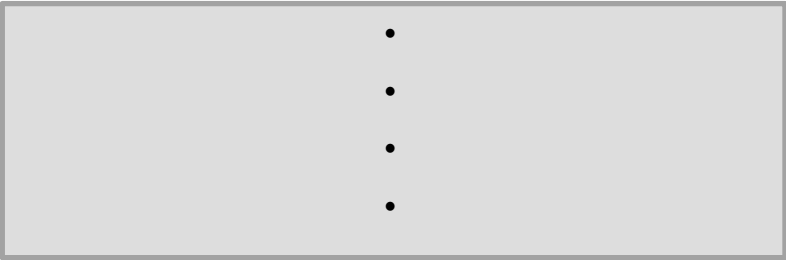
❑ Destrutor

- Pode ser definido utilizando o método **`__del__()`**. No entanto, ele não é tão necessário quanto no C++ uma vez que o Python usa um *Garbage Collector* para eliminar objetos sem referência automaticamente.

POO: Herança

- ❑ O conceito de herança em Python é o mesmo de outras linguagens, ou seja, **cria-se uma nova classe a partir de outra já pré-definida** no intuito de se **especializar** uma funcionalidade do software.
- ❑ Para realizar a herança, basta na definição da nova classe usar o nome da classe base entre parênteses.

class NovaClasse(ClasseBase):



-
-
-
-

POO: Herança

- ❑ Métodos da classe base podem ser sobrescritos na classe derivada;
- ❑ Caso deseja-se invocar métodos da classe base, pode-se utilizar a palavra-chave **super()**;
- ❑ Em Python o **polimorfismo** é suportado pelo interpretador, que irá ativar os métodos corretamente de acordo com o objeto.

Desafio

- ❑ Crie um sistema bancário similar ao que foi realizado na primeira parte do curso, de modo que ele possua as seguintes funcionalidades:
 1. Sistema de atendimento do banco, que deverá ser capaz de acionar as duas funcionalidades descritas abaixo.
 2. Sistema de atendimento para funcionários:
 1. Mudança de senha do gerente;
 2. Cadastro de novos clientes.
 3. Sistema de atendimento para clientes:
 1. Ver saldo.
 2. Realizar saque.
 3. Realizar depósito.