

# Programação em Python

## Estruturas

Prof. Daniel Di Domenico

[https://github.com/danidomenico/gex003\\_algprog](https://github.com/danidomenico/gex003_algprog)

# Estruturas em Python

- Estruturas, também chamados de **registros**;
- Em Python, são denominadas classes:
  - Conceito de classes será abordado em outras disciplinas do curso;
- São utilizadas para criar tipos de dados:
  - Tipos personalizados definidos pelo programador;
  - Combinam outros tipos, como **int**, **float**, **strings**...
    - Também é possível adicionar listas ou mesmo outras estruturas, porém requer maior entendimento de classes.

# Estruturas em Python

- Exemplo - estrutura Ponto:

```
#Estrutura Ponto, que possui 2 membros: x e y
```

```
class Ponto:
```

```
    x = 0.0
```

```
    y = 0.0
```

```
#Código principal...
```

```
p = Ponto() #Cria uma variável p do tipo Ponto
```

# Estruturas em Python

- Após declarar a variável com o tipo da estrutura, pode-se acessar os seus membros com o operador “.”:

```
#Estrutura Ponto, que possui 2 membros: x e y
class Ponto:
    x = 0.0
    y = 0.0

#Código principal...
p = Ponto()
p.x = 10.0 #Atribui valores para os...
p.y = 20.0 #membros x e y da variável p
print("Ponto:", p.x, p.y)
Ponto: 10.0 20.0
```

# Estruturas em Python

- Após declarar a variável com o tipo da estrutura, pode-se acessar os seus membros com o operador “.”:

```
#Estrutura Ponto, que possui 2 atributos: x e y
```

```
class Ponto:
```

```
    x = 0.0
```

```
    y = 0.0
```

```
#Código principal...
```

```
p = Ponto()
```

```
p.x = 10.0 #Atribui valores para os...
```

```
p.y = 20.0 #membros x e y da variável p
```

```
print("Ponto:", p.x, p.y)
```

```
Ponto: 10.0 20.0
```

**ATENÇÃO:** a execução do programa inicia do código principal (após as declarações de estruturas)

# Estruturas em Python

- Atenção ao **recuo (endentação!!!!)**;
- Não esquecer de recuar à direita:

```
class Ponto:  
    x = 0.0  
    y = 0.0
```

**CERTO!**

```
class Ponto:  
x = 0.0  
    y = 0.0
```

**ERRADO!**  
Deveria ter o  
mesmo recuo!

# Estruturas em Python

- As estruturas devem ser declaradas antes do código principal:

```
class Ponto:
```

```
    x = 0.0
```

```
    y = 0.0
```

```
#Código principal...
```

```
p = Ponto()
```

**CERTO!**

```
#Código principal...
```

```
p = Ponto()
```

```
class Ponto:
```

```
    x = 0.0
```

```
    y = 0.0
```

**ERRADO!**  
Estrutura declarada  
após o código  
principal.

# Estruturas em Python

- As estruturas devem ser declaradas antes das funções:

```
class Ponto:  
    x = 0.0  
    y = 0.0  
  
def imprime_ponto(p):  
    print(p.x, p.y)
```

**CERTO!**

```
def imprime_ponto(p):  
    print(p.x, p.y)  
  
class Ponto:  
    x = 0.0  
    y = 0.0
```

**ERRADO!**  
Estrutura declarada  
após à função.



# Estruturas em Python

- **Por que utilizar estruturas:**
  - Para armazenar diversas informações em uma mesma variável:
    - Pode-se, por exemplo, armazenar os dados de uma pessoa em uma única variável:

```
class Pessoa:  
    nome = ""  
    idade = 0  
    peso = 0  
    altura = 0.0
```

# Estruturas em Python

- Exemplo 1:
  - Programa para ler 5 pessoas e armazená-las em uma lista (vetor):

```
class Pessoa: #Estrutura Pessoa
    nome = ""
    idade = 0
    peso = 0
    altura = 0.0

#Código principal...
pessoas = []
for i in range(5):
    print("\nPessoa", i+1)
    pes = Pessoa()
    pes.nome = input("Informe o nome: ")
    pes.idade = int(input("Informe a idade: "))
    pes.peso = int(input("Informe o peso: "))
    pes.altura = float(input("Informe a altura: "))
    pessoas.append(pes) #Adiciona a pessoa na lista
```

# Estruturas em Python

- Exemplo 1 (com função):
  - Programa para ler 5 pessoas e armazená-las em uma lista (vetor):

```
class Pessoa: #Estrutura Pessoa
    nome = ""
    idade = 0
    peso = 0
    altura = 0.0

def ler_pessoa(numeroPessoa): #Função que lê os dados da pessoa e a retorna
    print("\\nPessoa", numeroPessoa)
    pes = Pessoa()
    pes.nome = input("Informe o nome: ")
    pes.idade = int(input("Informe a idade: "))
    pes.peso = int(input("Informe o peso: "))
    pes.altura = float(input("Informe a altura: "))
    return pes #retorna a variável do tipo Pessoa

#Código principal...
pessoas = []
for i in range(5):
    pessoas.append( ler_pessoa(i+1) ) #Adiciona a pessoa na lista
```

# Estruturas em Python

- Exemplo 2:
  - Função que imprime uma pessoa:

```
class Pessoa: #Estrutura Pessoa
    nome = ""
    idade = 0
    peso = 0
    altura = 0.0

def imprimir_pessoa(pessoa): #Função que imprime uma pessoa
    print()
    print("Nome:", pessoa.nome)
    print("Idade:", pessoa.idade)
    print("Peso:", pessoa.peso)
    print("Altura:", pessoa.altura)

#Código principal...
p = Pessoa()
p.nome = "Jhon Travolta"
imprimir_pessoa(p)
```

# Estruturas em Python

- 1) Baseado nos exemplos 1 e 2, faça um programa que após ler 5 pessoas, encontre e imprima a mais velha.
- 2) Faça um programa que possua uma estrutura Carro com os atributos: modelo, marca e ano de fabricação. Após, leia os valores para os 3 carros em uma função, retornando a lista com os 3 carros para o código principal.