PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
INTELIGÊNCIA ARTIFICIAL APLICADA
ADS1233
PROF. MSC. ANIBAL SANTOS JUKEMURA

PYTHON

Agenda:

- Módulos (continuação)
- Classes e objetos
- Classes e objetos: __init__
- Classes e objetos: exemplos
- Exercício em sala

PUC GOIÁS

- Módulos

- Se um programa se torna maior, é uma boa prática dividi-lo em arquivos menores, para facilitar a manutenção. Também é preferível usar um arquivo separado para uma função que você escreveria em vários programas diferentes, para não copiar a definição de função em cada um deles.

- Tal arquivo é chamado de módulo; definições de um módulo podem ser importadas para outros módulos, ou para o módulo principal (a coleção de variáveis a que você tem acesso num script executado como um programa e no modo calculadora).

PUC GOIÁS

- Módulos

```python
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()

def fib2(n):
    result = []
    a, b = 0, 1
    while a < n:
        result.append(a)
        a, b = b, a+b
    return result
```

Fibonacci.py

```python
import fibonacci as fib

fib.fib(1000)
print(fib.fib2(100))
fib.__name__
```

main.py

PUC GOIÁS

- Classes e objetos
- Sintaxe:

```
class ClassName:
    <statement-1>
    .
    .
    .
    <statement-N>
```

Cláusula self: faz referência a classe que foi criada, ou seja, faz referência ao próprio controle.

PUC GOIÁS

- Classes e objetos
- Exemplo 01:

```python
class calculadora:
    def soma(self,a,b):
        return a+b
    def multi(self, a,b):
        return a*b
    def subtr(self, a,b):
        return a-b
    def div(self, a,b):
        if b!=0:
            return a/b
        else:
            return 0
```

calculadora.py

```python
import calculadora as Calc
calc = Calc.calculadora()
print ("Soma: ", calc.soma(3,4))
print ("MultiplicacaO: ", calc.multi(3,4))
```

main.py

PUC GOIÁS

- Classes e objetos
- Exemplo 02: __init__

```python
class aluno:
    def __init__(self,nome, mat):
        self.nome=nome
        self.mat=mat
```

aluno.py

```python
import aluno

objeto=aluno.aluno('Goku')
print(objeto.nome)
print(objeto.mat)

objeto2=aluno.aluno('Japa',1000)
print(objeto2.nome)
print(objeto2.mat)
```

main.py

PUC GOIÁS

EXEMPLOS

1. Faça um projeto em python que possua uma classe Aluno com nome e matrícula. Guarde 5 alunos em uma lista e imprima-a.

```python
class Aluno:
    def __init__(self,nome,mat):
        self.nome = nome
        self.mat = mat
```

```python
import aluno
sala = []
for i in range (5):
    nome = input(f'Digite o nome {i+1}: ')
    mat = int(input(f'Digite a matricula {i+1} : '))
    objeto = aluno.Aluno(nome, mat)
    sala.append(objeto)

for i in range (5):
    print('Nome      : ', sala[i].nome)
    print('Matricula: ', sala[i].mat)
```

PUC GOIÁS

EXEMPLOS

2. Crie uma classe que modele um quadrado:

a. Atributos: Tamanho do lado
b. Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área;
c. No programa principal, crie uma lista com três quadrados fornecidos pelo usuário, altere o lado do segundo quadrado para o valor 10 (imprima esse lado), e por fim, imprima suas respectivas áreas.

```python
class quadrado:
    def __init__(self,lado):
        self.lado=lado

    def alteraLado(self,lado):
        self.lado = lado

    def getLado(self):
        return self.lado

    def area(self):
        return self.lado * self.lado
```

```python
import quadrado as quad

figuras = []
for i in range(3):
    lado = int(input(f'Digite o lado {i}: '))
    fig = quad.quadrado(lado)
    figuras.append(fig)

figuras[1].lado = 10
print('Nova area da Figura 2: ', figuras[1].lado)

for i in range(3):
    print(f'Area {i}: ', figuras[i].area())
```

PUC GOIÁS

EXERCÍCIO EM SALA – PARA ENTREGA

Crie uma classe para implementar uma conta corrente. A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo. Os métodos são os seguintes: alterarNome, depósito e saque; No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.

PUC GOIÁS

BIBLIOGRAFIA COMPLEMENTAR

- GRUS, J. **Data Science do zero: noções fundamentais com Python**. Rio de Janeiro: Alta Books, 2021.

- https://www.pythontutorial.net/