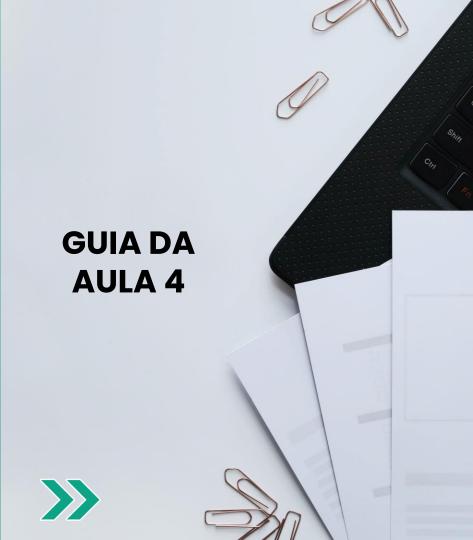


Python para análise de dados





PROGRAMAÇÃO ORIENTADA A OBJETOS







Aplique o conceito de herança



Manipulação



Acompanhe aqui os temas que serão tratados na videoaula





Definição

Uma especialização da classe.

```
class NomeClasse(object):
  def ___init____(self, params):
   . . .
class NomeClasseEspecializada(NomeClasse):
  def init (self, params): super(). init (self, params)
   self.atributo3 = ...
   self.atributo4 = ...
def metodo3(self, params):
 . . .
def metodo4(self, params):
 . . .
```





Repetindo a definição da classe Pessoa

```
from time import sleep
class Pessoa(object):
  def init (self, nome: str, idade: int, documento: str=None):
    self.nome = nome
    self.idade = idade
    self.documento = documento
  def dormir(self, horas: int) -> None: for hora in
   range (1, horas+1):
     print(f'Dormindo por {hora} horas') sleep(1)
  def falar(self, texto: str) -> None: print(texto)
  def str (self) -> str:
    return f'{self.nome}, {self.idade} anos e ' + \
         f'documento numero {self.documento}'
```



Criando a classe Universidade



```
class Universidade(object):

def _init_(self, nome: str): self.nome
= nome
```

Especializando a classe Pessoa na classe Estudante

```
In []:
    class Estudante(Pessoa):

    def _init_( self, nome: str,
        idade: int,
        documento: str, universidade:
        Universidade
    ):

    super()._init_(nome=nome, idade=idade, documento=documento)
    self.universidade = universidade
```



Manipulação



```
usp = Universidade(
            nome='Universidade de
             São Paulo'
          andre = Estudante( nome='Andre Perez', idade=30,
            documento='123', universidade=usp
         print(andre)
In [ ]:
         print(andre.universidade.nome)
```

