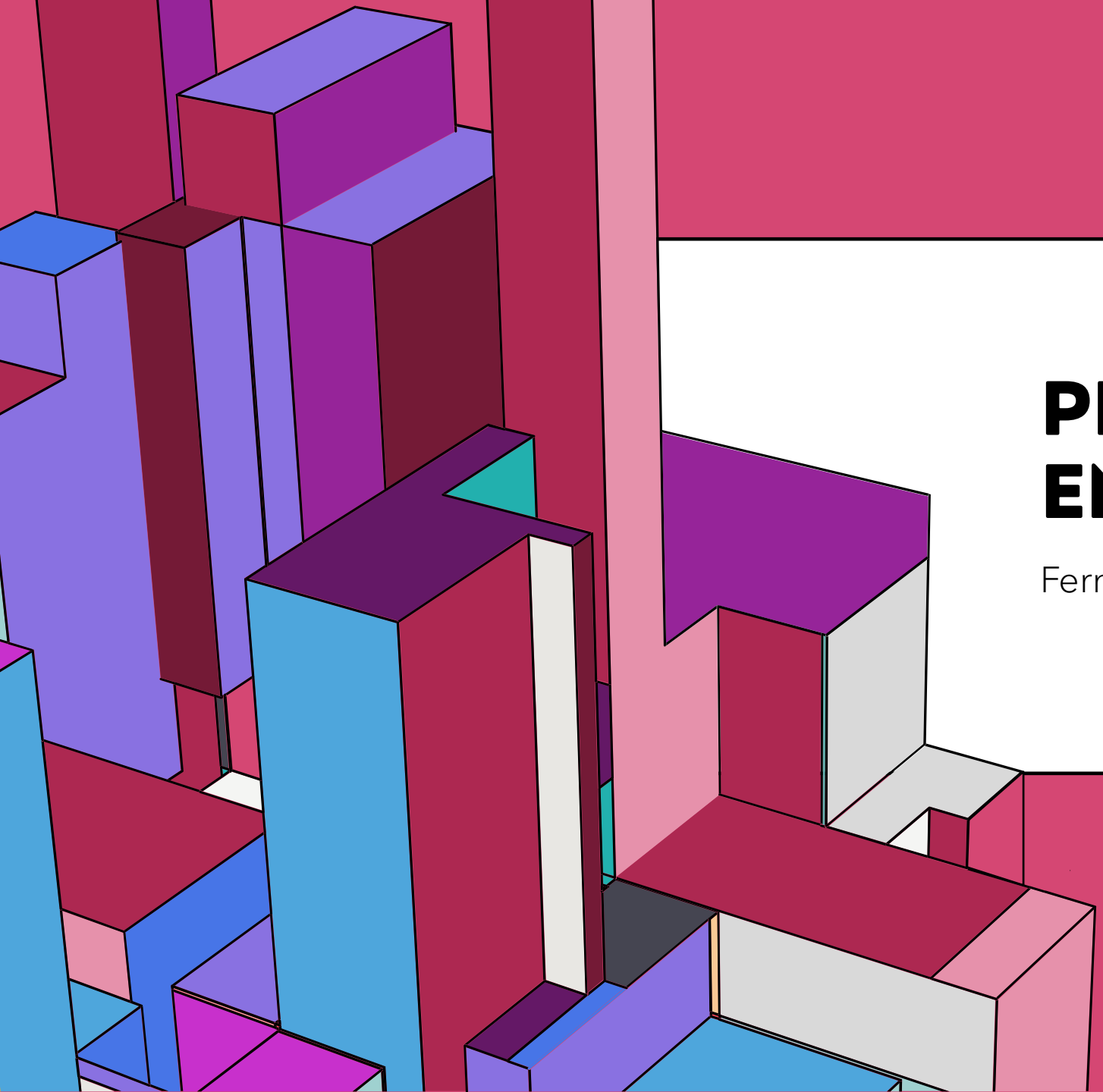


PROGRAMAÇÃO EM PYTHON

Fernando F. Santos

python

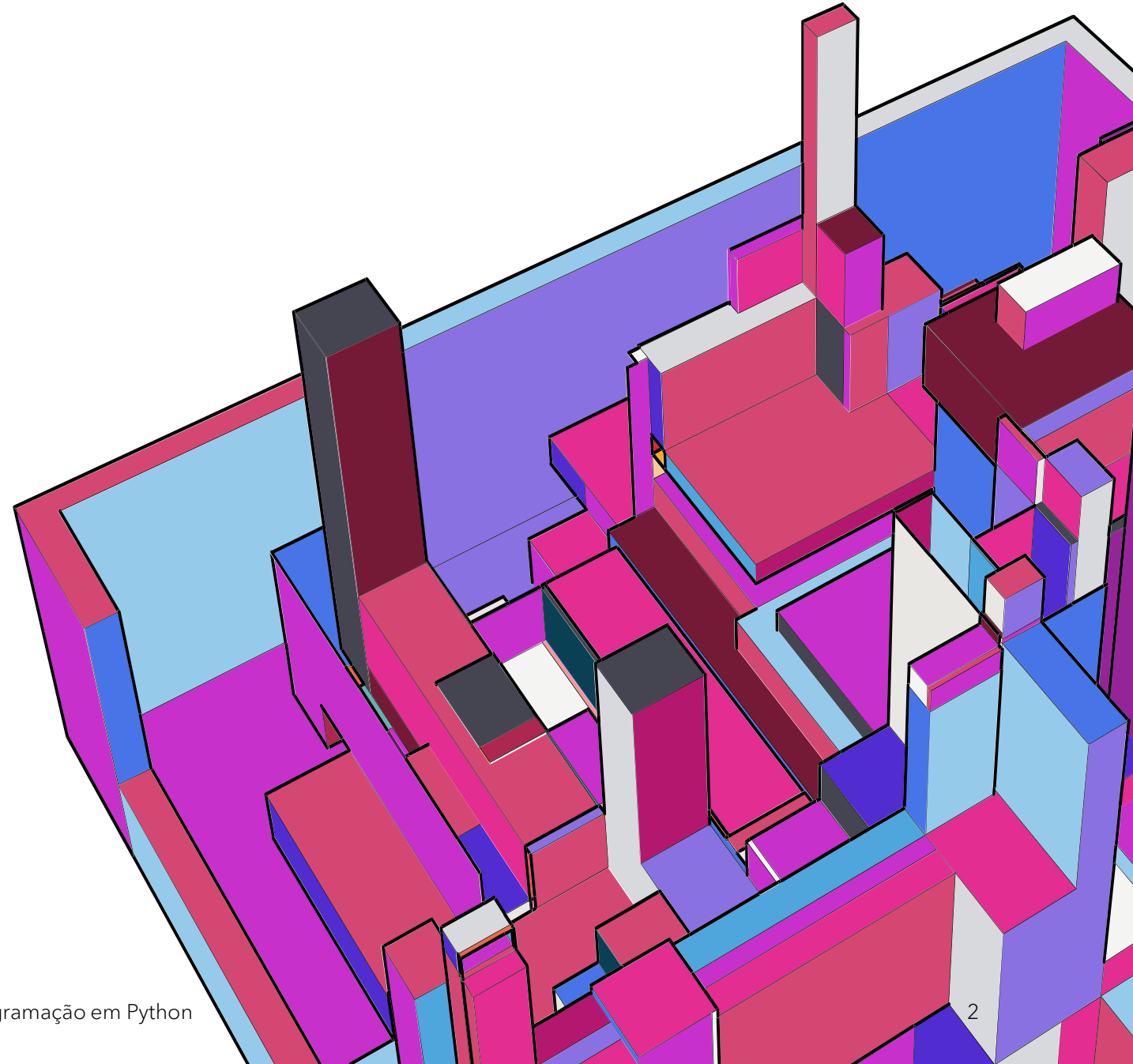
SENAI

AULA 12

15/03/2023

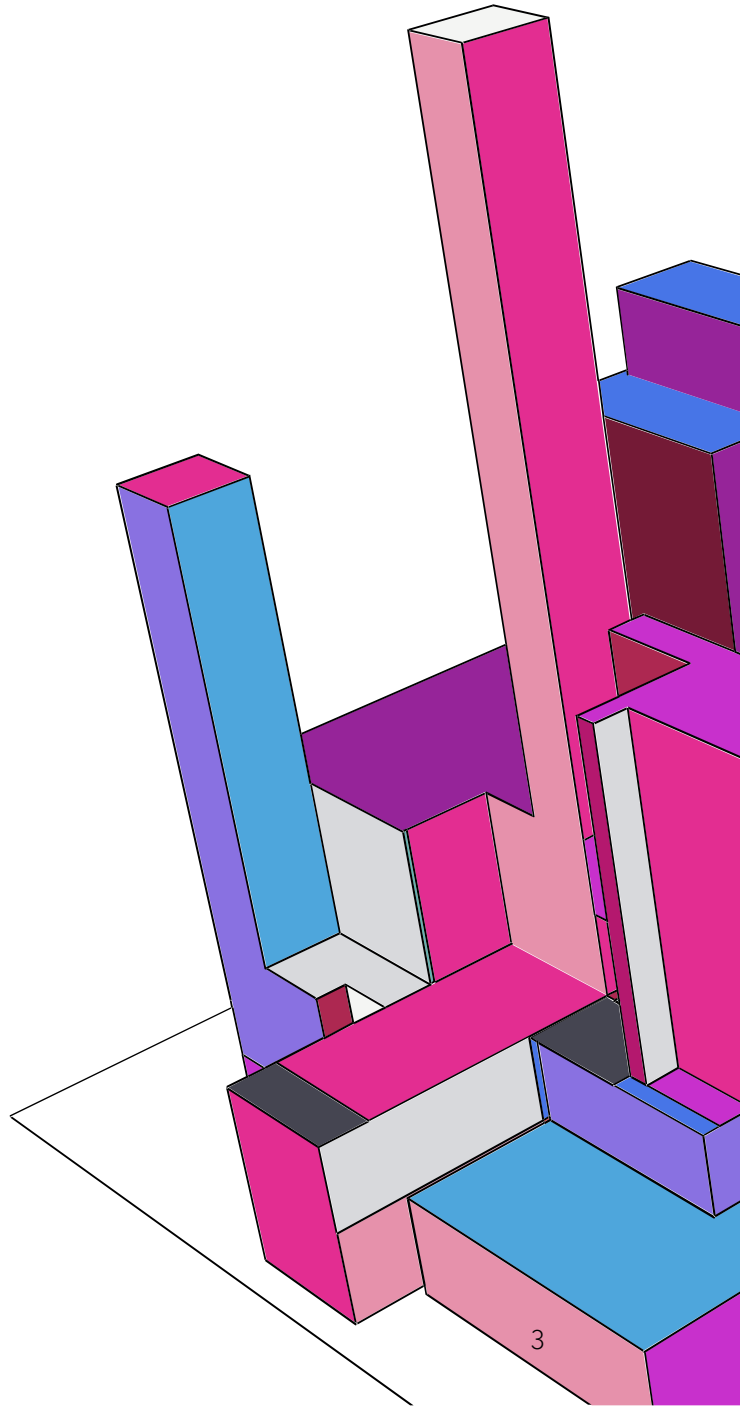
ORIENTAÇÃO A OBJETO

A programação orientada à objetos tem suas origens na década de 1960, mas somente em meados da década de 1980 ela se tornou no principal paradigma de programação utilizado na criação de software.



COMO FUNCIONA

Na programação orientada à objetos o foco é na criação de objetos que contem tanto os dados quanto as funcionalidades. Em geral, a definição de cada objeto corresponde a algum objeto ou conceito no mundo real e as funções que operam sobre tal objeto correspondem as formas que os objetos reais interagem. Na programação orientada à objetos o foco é na criação de objetos que contem tanto os dados quanto as funcionalidades. Em geral, a definição de cada objeto corresponde a algum objeto ou conceito no mundo real e as funções que operam sobre tal objeto correspondem as formas que os objetos reais interagem.



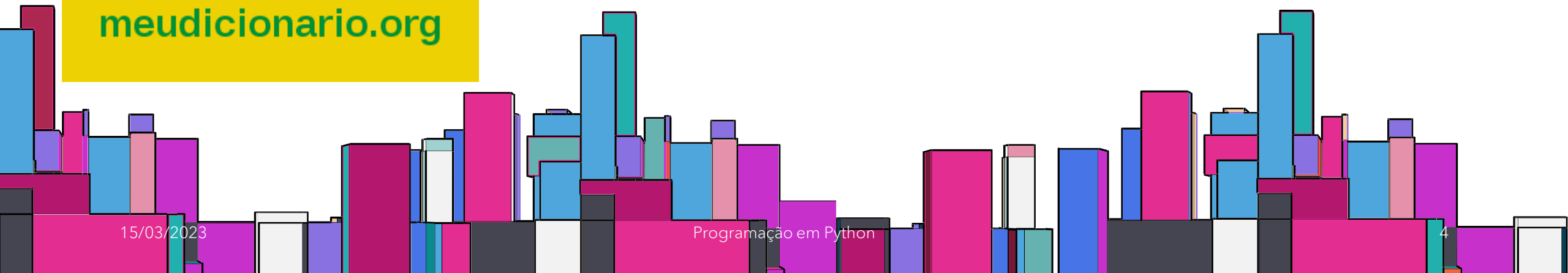
OBJETO

objeto

1. coisa material
2. tudo o que perceptível pelos sentidos
3. aquilo de que se trata; assunto, matéria, questão, tema
4. fim, propósito, finalidade, objetivo, fito

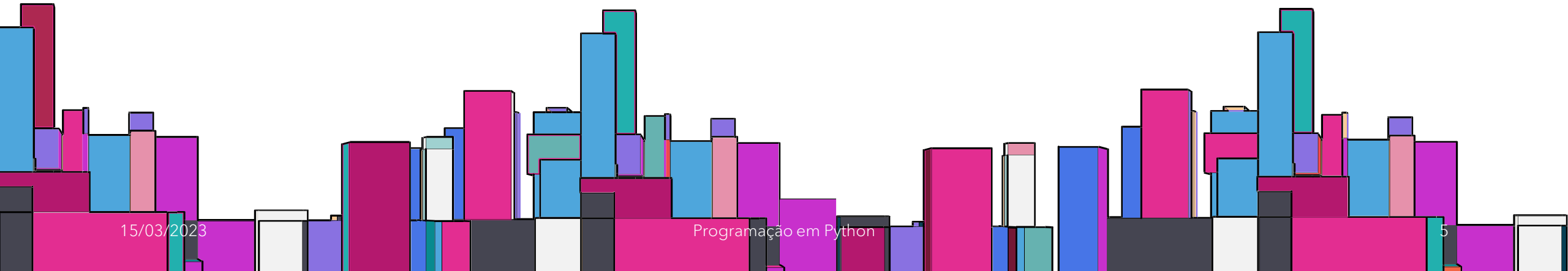
meudicionario.org

Quando ainda éramos bebês interagíamos com objetos (brinquedos infantis). Bebês aprendem que certos objetos fazem certas coisas: Sinos tocam, botões são pressionados e alavancas são puxadas. A definição de objeto em desenvolvimento de software não é muito diferente.



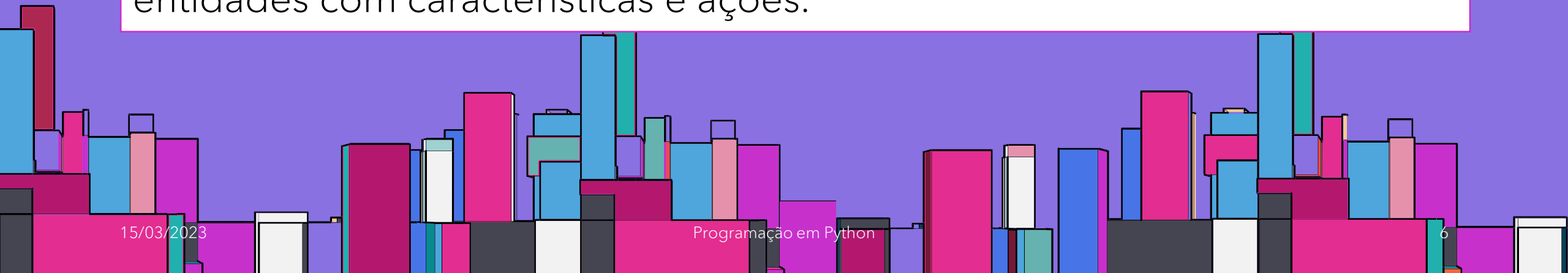
OBJETO NA VISÃO DA PROGRAMAÇÃO

Objetos de software não são coisas tangíveis, não tocamos ou sentimos, mas são modelos de algo que pode realizar certas ações.
No desenvolvimento de software objeto é uma coleção de dados que possui comportamentos associados.
Um software é uma coleção de objetos que se comunicam entre si, dizendo uns aos outros o que devem fazer



OBJETOS DE SOFTWARE

Objetos de software, assim como objetos do mundo real possuem características (propriedades, atributos) e ações (procedimentos, métodos).
Os **atributos** são estruturas de dados que armazenam informações sobre o objeto.
Os **métodos** são responsáveis pela manipulação dos atributos, são funções associadas ao objeto que descrevem como o objeto se comporta.
Resumindo, objetos são abstrações computacionais que representam entidades com características e ações.

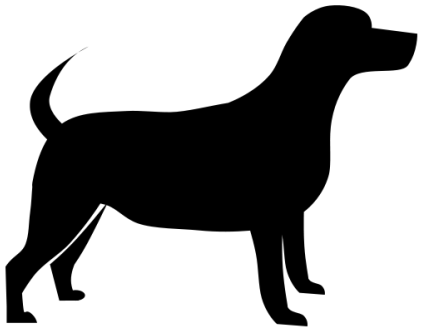


EXEMPLO

Atributos: Cor, Marca, Modelo, Ano, Capacidade do tanque.

Métodos: Acelerar, frear, virar à esquerda, virar à direita.

Atributos: Raça, idade, peso, cor do pelo.



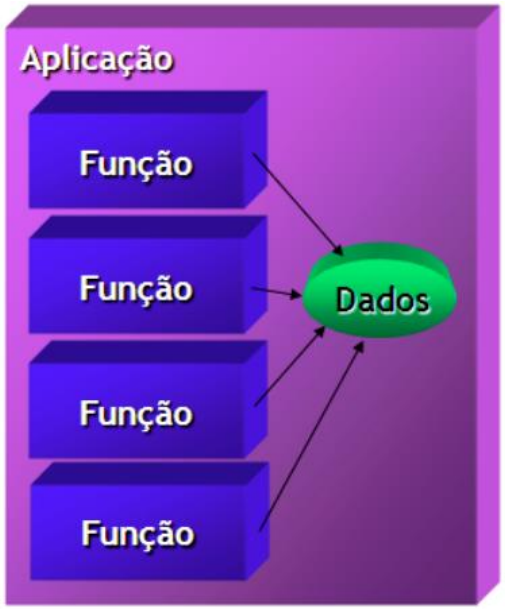
Métodos: Latir, andar, dormir, comer, morder.

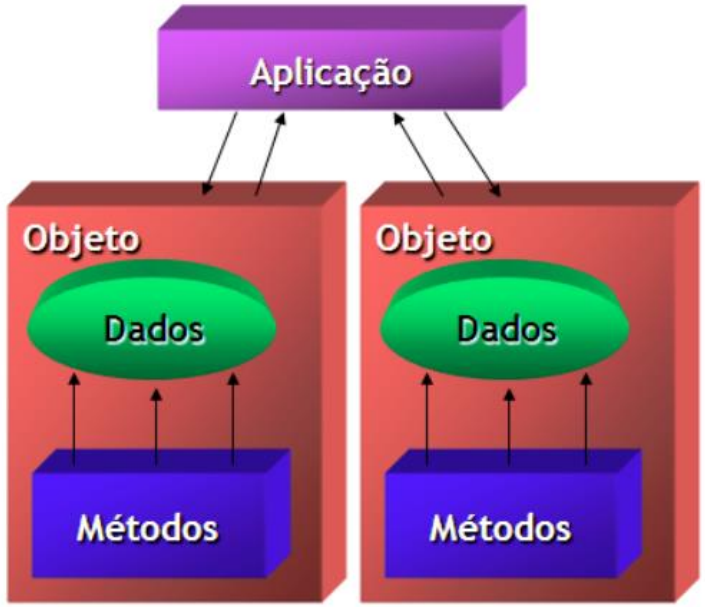
PARADIGMA

Forma de abordar um problema

Paradigma de programação estruturada: Os programadores abstraem o programa como uma sequência de funções executadas de modo empilhado.

Paradigma de programação orientada a objetos: Programadores abstraem o programa como uma coleção de objetos que interagem entre si.





CLASSE

É a estrutura básica do paradigma de orientação a objetos, que representa o tipo do objeto, um modelo a partir do qual os objetos serão criados.
Classes são abstrações utilizadas para representar um conjunto de objetos com características e comportamentos idênticos.
Uma classe pode ser vista como uma fábrica de objetos.

Classe: Veiculo

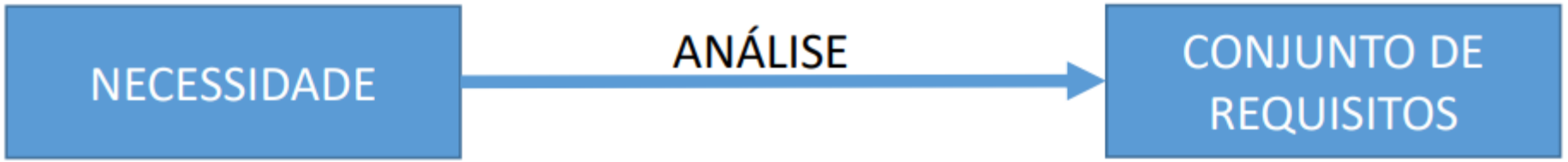
Objetos da classe Veiculo

ORIENTAÇÃO A OBJETOS

Orientado = direcionado para.
Sendo assim, orientado a objeto é praticamente direcionado para modelos de objetos.
Orientação a objetos é uma técnica para modelar sistemas complexos, descrevendo uma coleção de objetos interagindo através de seus dados e comportamentos.
Dividimos em quatro conceitos que são: **Análise Orientada a Objetos**, **Design Orientado a Objetos** e **Programação Orientada a Objetos**. Todos são estágios do desenvolvimento de software. Chamá-los de "orientados a objetos" especifica o estilo de desenvolvimento que está sendo seguido.

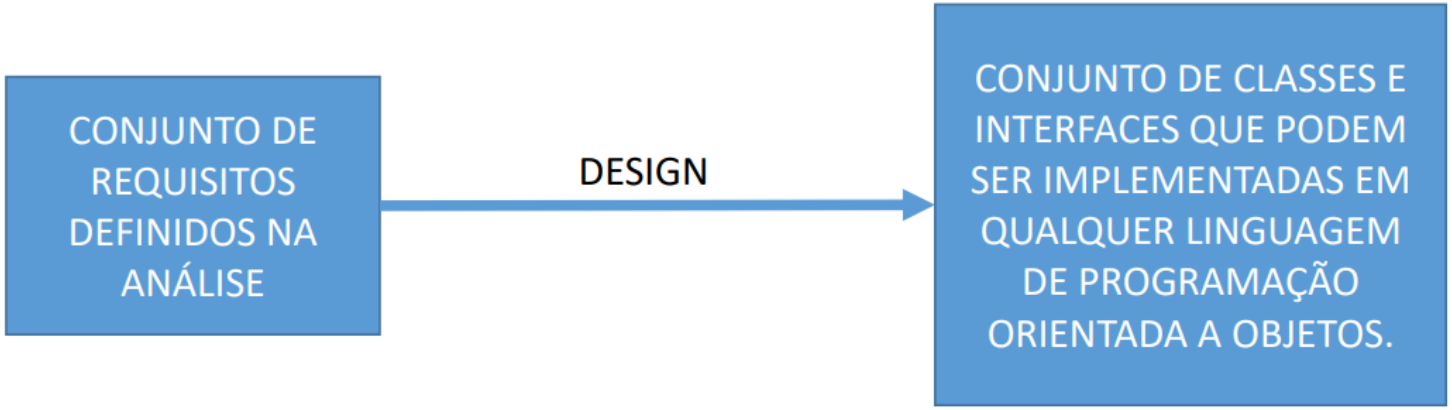
ANÁLISE ORIENTADA A OBJETOS

É o processo de olhar para um problema, sistema ou tarefa (que alguém quer transformar em um aplicativo) e identificar os objetos e as interações entre esses objetos.
A produto final do estágio de análise é um conjunto de requisitos.
Em desenvolvimento de softwares, o estágio da análise inclui entrevistar clientes, estudar seus processos e eliminar possibilidades.



DESIGN ORIENTADO A OBJETOS

O design orientado a objetos é o processo de converter os requisitos em uma especificação de implementação. O designer deve nomear os objetos, definir os comportamentos e especificar formalmente quais objetos podem ativar comportamentos específicos em outros objetos. Esse estágio é responsável por definir como as coisas devem ser feitas.



PROGRAMAÇÃO ORIENTADA A OBJETOS

É o processo de converter o que foi definido pelo design em um programa que faz o que originalmente foi solicitado. É uma técnica de programação que organiza nossos programas em classes e objetos em vez de apenas funções.

Python não impõe a programação orientada a objetos como o principal paradigma de programação.

ORIENTAÇÃO A OBJETOS NO PYTHON

Em Python, tudo é um objeto e pode ser manipulado como tal. Funções, classes, strings e até mesmo tipos são objetos. Todos têm um tipo, podem ser passados como argumento de função e podem ter métodos e propriedades. Nesse aspecto, Python é uma linguagem orientada a objetos. Entender os conceitos essenciais da programação orientada a objetos envolve compreender os conceitos de objetos, classes e métodos.

CRIANDO CLASSES NO PYTHON

- No Python, novos objetos são criados a partir das classes por meio de atribuição. O objeto é uma instância da classe que possui características próprias. As classes são derivadas da classe base denominada object. Veja como criamos um objeto:

Objeto = Classe()

É como se o objeto fosse uma variável e o tipo fosse a classe.

Toda instância de classe ou variável tem seu próprio endereço de memória ou sua identidade.

Os objetos, que são instâncias de classes interagem entre si para servir ao propósito de uma aplicação em

desenvolvimento.

CRIANDO CLASSES NO PYTHON

- Para criar uma classe em Python usamos a palavra reservada class.

```
Class nome_classe:
    variavel = valor
    #...
    variavel_n = valor
    def metodo(self, ...arg):
        #...
    def metodo(self, ...arg):
        #..
```

O primeiro argumento de um método é o self. Esta variável representa o próprio objeto. O nome self é uma convenção, podendo ser trocado por outro nome qualquer, porém **é considerado como boa prática manter este nome**

EXEMPLO

```
# Classe de Herói
class Heroi:
    #Definição do Atributos da classe:
    voa = False
    possui_arma = False
    lanca_teia = False
    frase_comum = ""
    #Definição dos métodos/;
    def falar(self):
        print(self.frase_comum)
    def detalhar(self):
        if self.voa:
            print("O herói voa.")
        if self.possui_arma:
            print("O herói possui arma.")
        if self.lanca_teia:
            print("O herói lança teia.")
```

```
#Programa Principal
homem_aranha = Heroi() #cria um objeto do tip
o Heroi
homem_aranha.lanca_teia = True
print(homem_aranha.voa)
print(homem_aranha.lanca_teia)
he_man = Heroi()  #cria um objeto do tipo Her
oi
he_man.possui_arma = True
he_man.lanca_teia = False
he_man.voa = False
he_man.frase_comum = "Eu tenho a força"
he_man.falar()
homem_aranha.detalhar()
he_man.detalhar()
```

EXEMPLO 2 (COM SEPARAÇÃO EM ARQUIVOS)

heroi.py

```python
# Classe de Herói
class Heroi:
    #Definição do Atributos da classe:
    voa = False
    possui_arma = False
    lanca_teia = False
    frase_comum = ""
    #Definição dos métodos/;
    def falar(self):
        print(self.frase_comum)
    def detalhar(self):
        if self.voa:
            print("O herói voa.")
        if self.possui_arma:
            print("O herói possui arma.")
        if self.lanca_teia:
            print("O herói lança teia.")
```

sistema.py

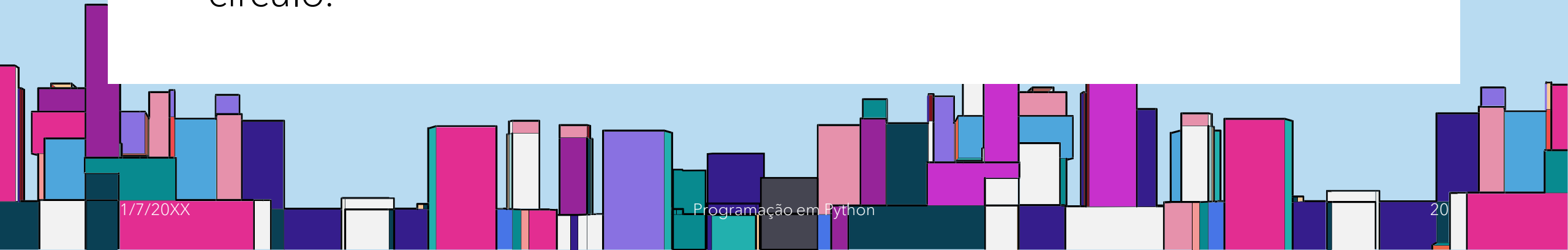
```python
#Programa Principal
from heroi import Heroi #imprta o arquivo
homem_aranha = Heroi() #cria um objeto do tipo
Heroi
homem_aranha.lanca_teia = True
print(homem_aranha.voa)
print(homem_aranha.lanca_teia)
he_man = Heroi()  #cria um objeto do tipo Heroi
he_man.possui_arma = True
he_man.lanca_teia = False
he_man.voa = False
he_man.frase_comum = "Eu tenho a força"
he_man.falar()
homem_aranha.detalhar()
he_man.detalhar()
```

EXERCÍCIOS

1. Cite 4 atributos e 3 métodos de um aluno
2. Seguindo os estudo de orientação a objeto quais as classes para um sistema de uma mercearia?
3. Escolha 3 classes da atividade anterior e crie seus atributos e métodos.
4. Faça um esboço(word) que defina classes, atributos e métodos para um controle de estoque.

EXERCÍCIOS

5. Escreva uma definição para uma classe denominada Circle, com os atributos center e radius, onde center é um objeto Point e radius é um número.
   - Instancie um objeto Circle, que represente um círculo com o centro em 150, 100 e raio 75.
   - Escreva uma função denominada point_in_circle, que tome um Circle e um Point e retorne True, se o ponto estiver dentro ou no limite do círculo.



EXERCÍCIOS

10 . Escreva uma classe cujos objetos representam alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, 2 notas de prova e 1 nota de trabalho. Escreva os seguintes métodos para esta classe:

media	calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
final	calcula quanto o aluno precisa para a prova final (retorna zero se ele não for para a final)

EXERCÍCIOS

11. Escreva uma classe Data cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

construtor	define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001
compara	recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne: • 0 se as datas forem iguais; • 1 se a data corrente for maior que a do parâmetro; • -1 se a data do parâmetro for maior que a corrente.
getDia	retorna o dia da data
getMes	retorna o mês da data
getMesExtenso	retorna o mês da data corrente por extenso
getAno	retorna o ano da data