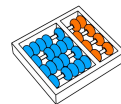


INTRODUÇÃO À PYTHON

Capacitação Profissional em Tecnologias de Inteligência Artificial

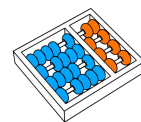
Allan M. de Souza

Instituto de Computação - Universidade Estadual Campinas



O QUE É PYTHON?





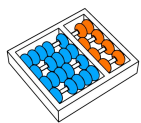
PYTHON



- Python é uma linguagem de programação de **alto nível**, poderosa, simples, e multi-paradigma
- Computadores executam programas em linguagens de **baixo nível** (“linguagem de máquina” ou “linguagens assembly”)
- Programas escritos em linguagens de alto nível precisam ser processados antes de serem executados



PYTHON



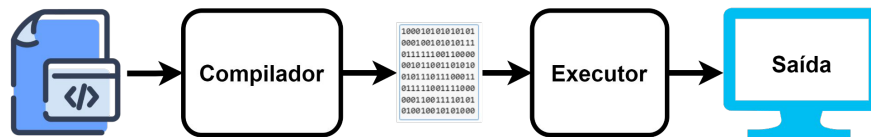
- Dois tipos de programas processam linguagens de alto nível
 - Interpretadores
 - Compiladores
- **Interpretador:** : lê um programa escrito em linguagem de alto nível e o executa passo a passo, interpretando cada um dos comandos individualmente.



PYTHON

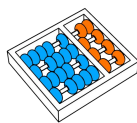
— — —

- **Compilador:** lê o programa e o traduz completamente antes que o programa comece a rodar.



- O programa traduzido é chamado de código objeto ou executável.
- O Python usa **ambos** os processos, mas ela é **em geral considerada uma linguagem interpretada**.

PYTHON



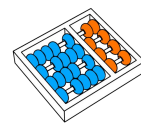
- Existem duas maneiras de usar o interpretador
 - linha de comando (“shell mode”)
 - script (“program mode”).
- **Linha de comando:** você digita comandos em Python e o interpretador mostra o resultado
- **Script:** você pode escrever um programa inteiro em um arquivo e usar o interpretador para executar o conteúdo do arquivo como um todo.

```
print("Programa em python para somar os números 2 e 3")  
print(2 + 3)
```



PYTHON

— — —



Shell mode

```
$ python
```

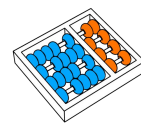
```
>>> print("programa em python para somar os números 2 e 3")
Programa em python para somar os números 2 e 3
>>> print(2 + 3)
5
```



Script mode

```
$ python programa1.py
```

```
Programa em python para somar os números 2 e 3
5
```



IDEs PARA DESENVOLVIMENTO

Um ambiente de desenvolvimento integrado (IDE) é uma aplicação de software que ajuda os programadores a desenvolver código de software de maneira eficiente

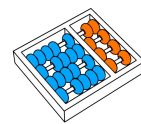
- Jupyter Notebook
- PyCharm
- Spyder
- Visual Studio
- Editor de texto
- ...



Visual Studio Code



PyCharm

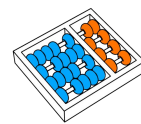


ESTRUTURA BÁSICA DE UM PROGRAMA

- Um **programa** Python é uma sequência de comandos que serão executados pelo interpretador Python.

```
comando 1  
comando 2  
...  
comando n
```

- O programa deve ter **um comando por linha**. Os comandos serão executados nesta ordem, **de cima para baixo**, um por vez



ESTRUTURA BÁSICA DE UM PROGRAMA

— — —

- Estrutura correta

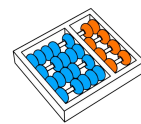
```
print("Ola pessoal")  
print("Vamos aprender python")
```



- Estrutura errada (2 comandos na mesma linha)

```
print("Ola pessoal") print("Vamos aprender python")
```





ESTRUTURA BÁSICA DE UM PROGRAMA

— — —

- Estrutura correta

```
print("Ola pessoal")  
print("Vamos aprender python")
```



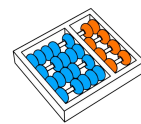
- Estrutura errada (2 comandos na mesma linha)

```
print("Ola pessoal") print("Vamos aprender python")
```



```
print("Ola pessoal"); print("Vamos aprender python")
```





ESTRUTURA BÁSICA DE UM PROGRAMA

— — —

- Indentação correta

```
print("Ola pessoal")  
    print("Vamos aprender python")
```



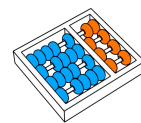
- Indentação incorreta

```
print("Ola pessoal")  
    print("Vamos aprender python")
```



- Python não faz uso de {} para controlar o escopo de função e variáveis
- Portanto a indentação **deve ser respeitada**

OBTENDO PYTHON



Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [»»» Learn More](#)

ANACONDA. Products Pricing Solutions Resources Partners Blog Company Contact Sales

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

[Download](#)

For Windows
Python 3.9 • 64-Bit Graphical Installer • 621 MB

Get Additional Installers

Windows Apple Linux

Open Source

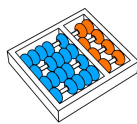
Access the open-source software you need for projects in any field, from data visualization to robotics.

User-friendly

With our intuitive platform, you can easily search and install packages and create, load, and switch between environments.

Trusted

Our securely hosted packages and artifacts are methodically tested and regularly updated.

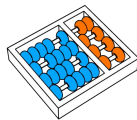


OBJETOS

- Um programa executa comandos para manipular informações/dados.
- Qualquer dado em Python é um objeto, que é de um certo tipo específico.
- O tipo de um objeto especifica quais operações podem ser realizadas sobre o objeto.
- Por exemplo, o número 5 é representado com um objeto 5 do tipo `int` em Python.

OBJETOS

— — —

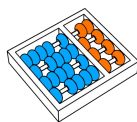


```
print(type("Olá pessoal"))  
print(type(5))  
print(type("5"))
```

```
<class 'str'>  
<class 'int'>  
<class 'str'>
```

- Olá pessoal é uma **string** do tipo **str**
- Enquanto 5 é um inteiro do tipo **int**
- Por outro lado, 5 é um número inteiro, mas como entre aspas é uma **string**.

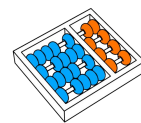
VARIÁVEIS



- Variáveis são uma forma de se associar um nome dado pelo programador com um objeto.
- No exemplo abaixo associamos os nomes altura, largura e a com os valores 10, 3, e 29, respectivamente.
- O operando de atribuição em python é representado por `=`. Ele associa a variável do lado esquerdo do comando com o objeto do lado direito do comando.

```
altura = 10
largura = 3
a = 29
a = 'a'
```

- Variáveis em python são fracamente tipadas, ou seja o valor de uma variável pode ser alterado independentemente do tipo anteriormente atribuído

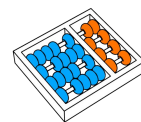


VARIÁVEIS - DIAGRAMA DE REFERÊNCIA

— — —

Diagrama de referência mostra o estado de cada variável em um **instante de tempo particular**

```
altura = 10  
largura = 3  
a = 29  
a = 'a'
```



VARIÁVEIS - DIAGRAMA DE REFERÊNCIA

— — —

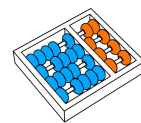
Diagrama de referência mostra o estado de cada variável em um **instante de tempo particular**

altura



10

```
altura = 10  
largura = 3  
a = 29  
a = 'a'
```



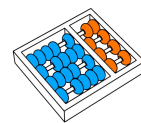
VARIÁVEIS - DIAGRAMA DE REFERÊNCIA

— — —

Diagrama de referência mostra o estado de cada variável em um **instante de tempo particular**



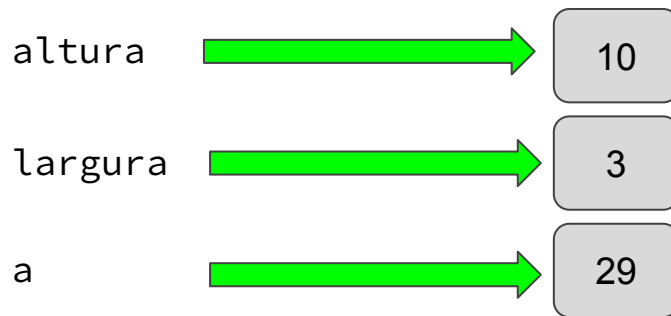
```
altura = 10  
largura = 3  
a = 29  
a = 'a'
```



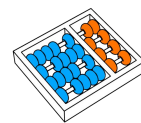
VARIÁVEIS - DIAGRAMA DE REFERÊNCIA

— — —

Diagrama de referência mostra o estado de cada variável em um **instante de tempo particular**



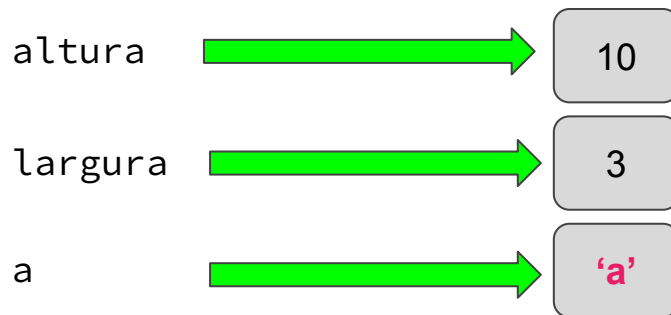
```
altura = 10  
largura = 3  
a = 29  
a = 'a'
```



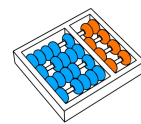
VARIÁVEIS - DIAGRAMA DE REFERÊNCIA

— — —

Diagrama de referência mostra o estado de cada variável em um **instante de tempo particular**



```
altura = 10
largura = 3
a = 29
a = 'a'
```



VARIÁVEIS - REGRAS PARA NOMES

- Deve começar com uma letra (maiúscula ou minúscula) ou underscore(_). Nunca pode começar com um número.
- Pode conter letras maiúsculas, minúsculas, números e subscrito.
- Não pode-se utilizar como parte do nome de uma variável: { (+ - * / \ n ; . , ?
- Não pode-se usar palavras reservadas: **for**, **while**, **class**, **if**, **else** ...
- Letras maiúsculas e minúsculas são diferentes: `c = 4` `C = 3`



`4rodas = 4`



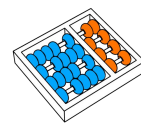
`dinheiro$ = 100`



`lista-itens = [a, b, c]`



`class = [a, b, c]`



TIPOS DE DADOS EM PYTHON

— — —

Numéricos: Integer, Float, Complex ($X + Yj$)

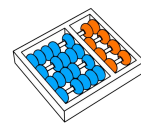
Text: String

Listas

Dicionários

Tuplas

Sets



LISTAS EM PYTHON

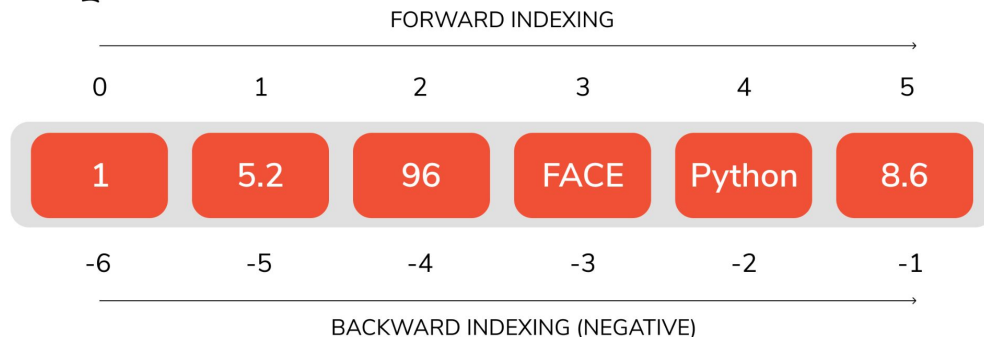
É uma estrutura de dados que armazena uma sequência de objetos **homogêneos** ou **heterogêneos**

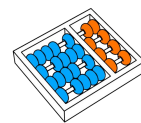
Podemos acessar os elementos da lista usando `lista[idx]`

Listas podem ser indexadas com índices **positivos** quanto com índices **negativos**

List = [1, 5.2, 96, "FACE", "Python", 8.6]

View of a list
↓



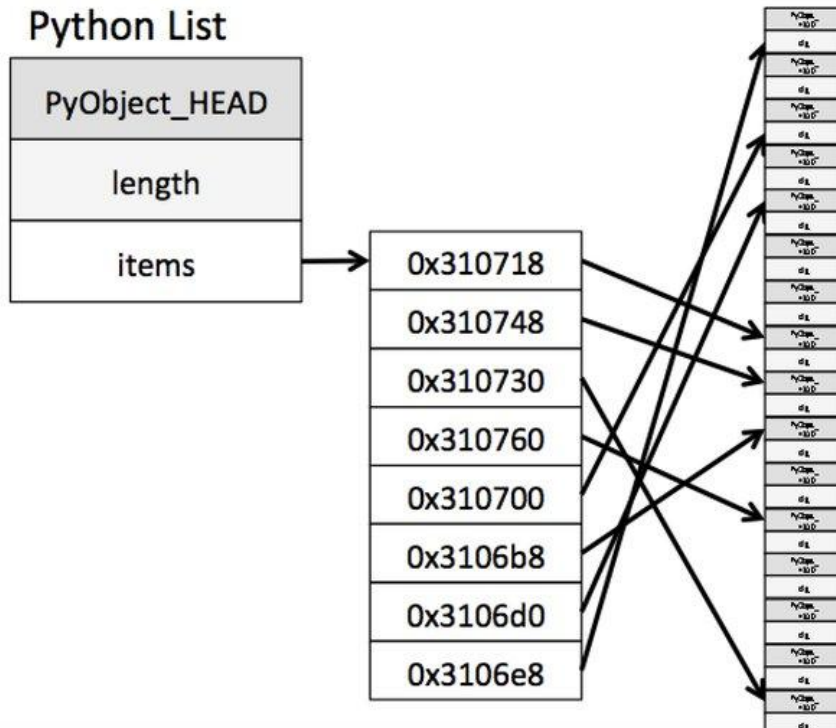


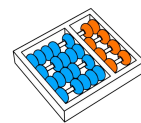
LISTAS EM PYTHON

Para permitir objetos heterogeneous python utiliza a seguinte estrutura

Os itens da lista são referências para endereços de memória onde o valor está armazenado

Maior flexibilidade porém menor performance



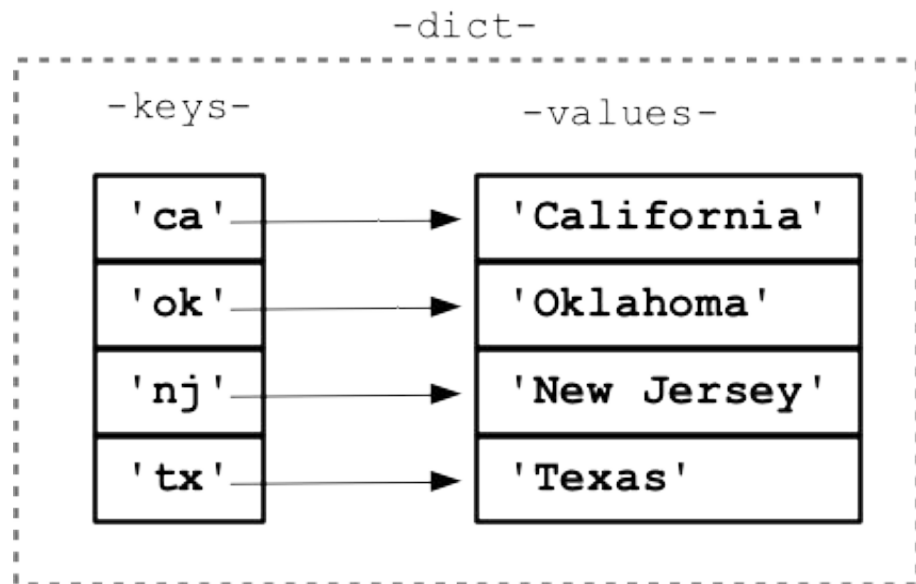


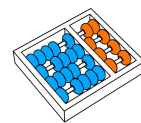
DICIONÁRIOS EM PYTHON

É uma estrutura de dados que armazena objetos <chave, valor>

Assim, os valores ou objetos podem ser referenciados pela chave

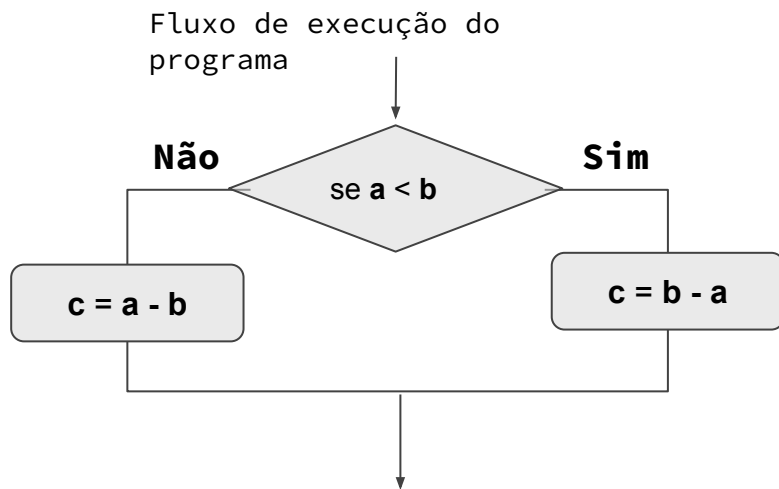
Diferente das listas onde são indexadas por **índices inteiros**





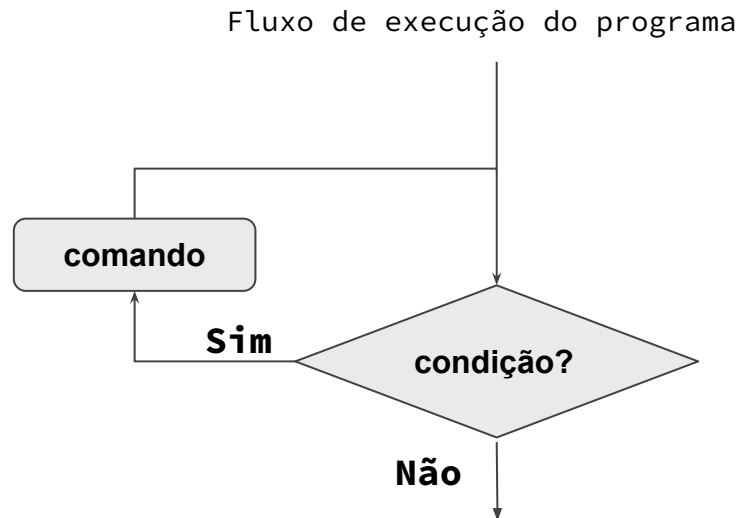
CONTROLE DE FLUXO

— — —



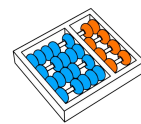
Condicionais

IF ELSE ELIF

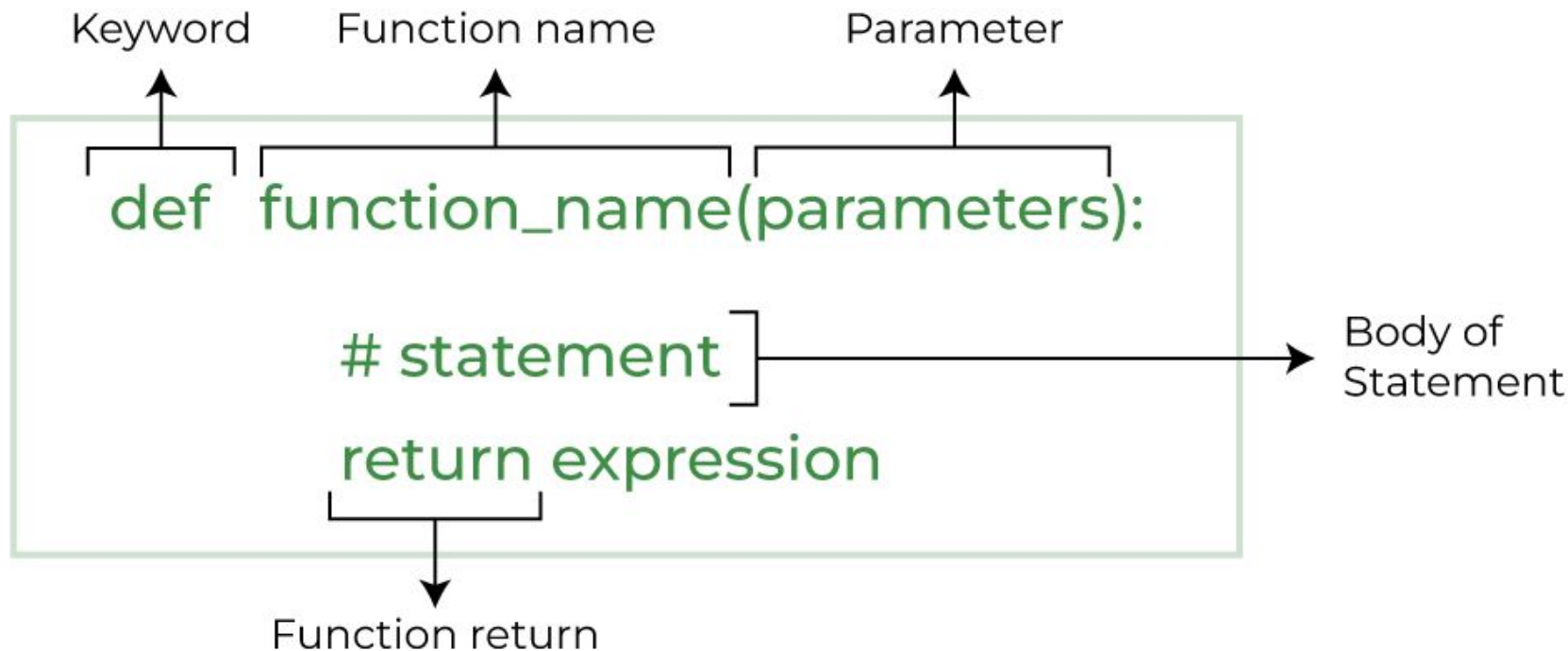


Repetição

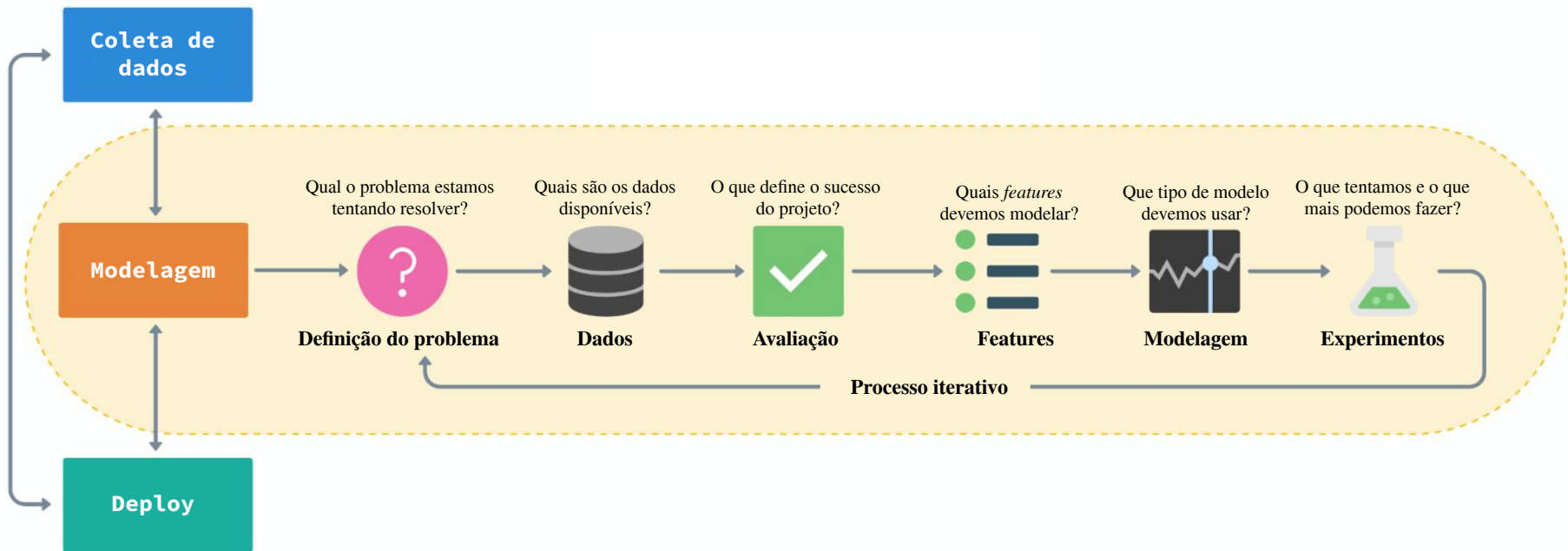
WHILE FOR

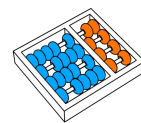


FUNÇÕES EM PYTHON

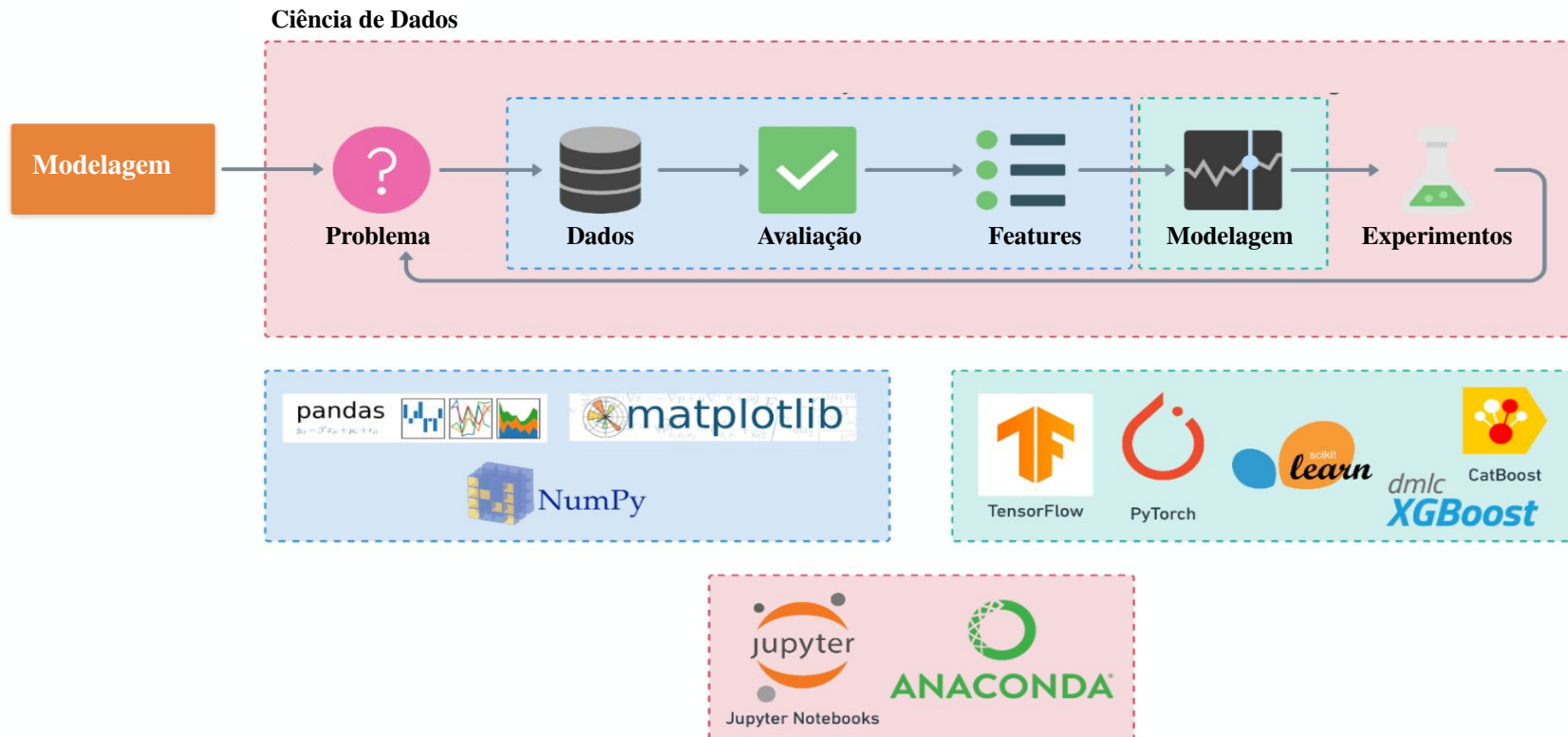


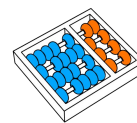
PYTHON PARA CIÊNCIA DE DADOS





PYTHON PARA CIÊNCIA DE DADOS





OBTENDO PACOTES AUXILIARES

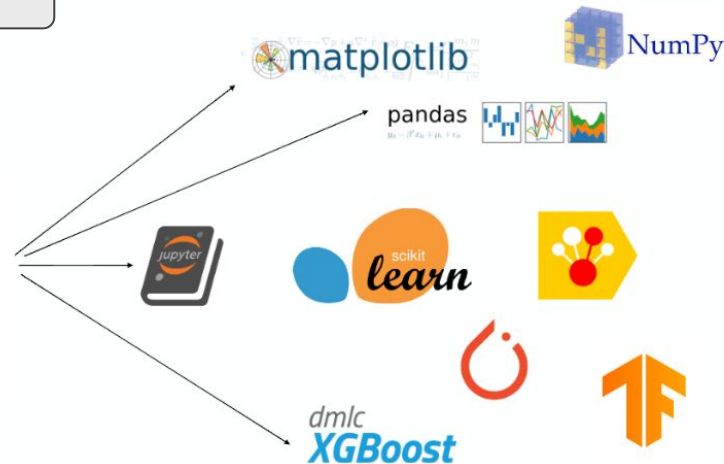
```
$ pip install <nome_pacote>
```



```
$ conda install <nome_pacote>
```



Ambiente
desenvolvimento



<pacotes>

INTRODUÇÃO À PYTHON

Próxima aula:

- Hands on com Python

Variáveis,

Operadores aritméticos,
relacionais e lógicos

Controle de fluxo

Operações com listas e
dicionários

Funções

