

Tecnólogo em Análise e Desenvolvimento de Sistemas

Desenvolvimento Web I



UNIVERSIDADE
CANDIDO
MENDES

Prof. Ricardo Tavares

ricardo.tavares@ucam-campos.br

■ Roadmap

- Introdução a linguagem de programação Python;
- Definição e atribuição de variáveis e constantes;
- Operadores matemáticos, lógicos e relacionais;
- Estruturas condicionais e de repetição;
- Estruturas de dados: listas, matrizes, tuplas e dicionários;
 - Compreensão de listas;
- Utilização de bibliotecas;
- Funções e Lambda;
- Modularização em pacotes;
- POO em Python;
 - Classes e objetos;
 - Atributos e métodos;
 - Heranças;

■ Roadmap

- Introdução a aplicações web;
- Introdução ao *framework* Django;
- Criação de projetos e apps;
- Arquitetura MVC/MVT;
- Utilizando banco de dados e Django ORM;
- Desenvolvimento de aplicação web (projeto final da disciplina);

■ Objetivo

- Desenvolver competências para criação de aplicações utilizando a linguagem de programação Python;
- Entender como utilizar o paradigma da orientação a objetos utilizando Python;
- Adquirir aptidões para desenvolvimento de aplicações web utilizando POO, Python e Django;
- Desenvolver uma *web application* completa utilizando Python, Django e MySQL/SQLite;
 - Alunos devem definir que tipo de aplicação quer desenvolver;
 - Será necessário a utilização de POO;
 - Será necessário a construção de diagrama de caso de uso e diagrama de classes;
 - Exemplos: Controle de biblioteca, controle de estoque, controle de folha de pagamento, construção de e-commerce (imóveis, supermercado, etc.);
 - Utilizar bootstrap para criação de front-end;

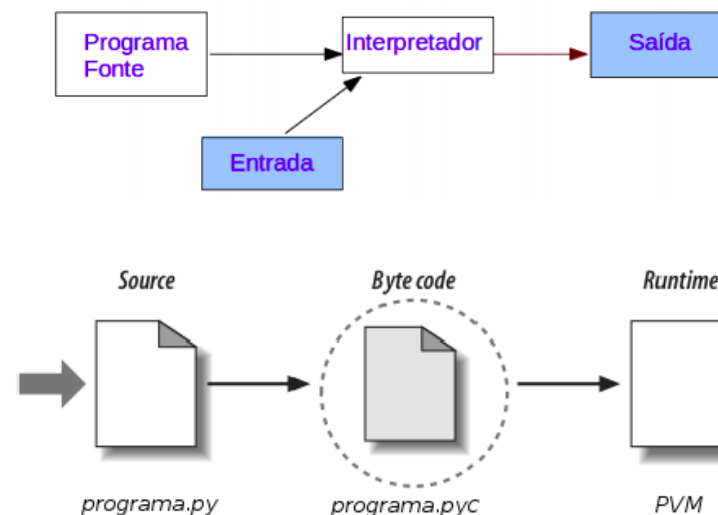
Introdução ao Python



■ Introdução a Linguagem de Programação Python

■ O que é o Python?

- Linguagem robusta, moderna e bem projetada:
 - Desenvolvimento de todo tipo de software: desktop, web, estatísticos, ciência de dados, machine learning, etc;
- Multiparadigma: Procedural e Orientada a Objetos;
- Multiplataforma e portátil: interpretada;
- Modular e escalável:
 - Permite a construção de módulos, pacotes, classes, API's, etc;
 - Biblioteca built-in extensa;
 - Permite a criação/utilização de bibliotecas desenvolvidas por terceiros, expandindo suas funcionalidades de forma prática e rápida;



■ Introdução a Linguagem de Programação Python

■ O que é o Python?

- Open Source e gratuita;
- Permite a utilização de vários gerenciadores de ambientes:
 - Anaconda e Virtual Env;
- Fácil utilização de paradigmas de desenvolvimento, como: TDD (Test Driven Development) e XP (Extreme Programming);
 - Possui biblioteca completa para elaboração de testes automatizados (Unit Test Framework);
- Desenvolvida sob Paradigma de Orientação a Objetos:
 - Tudo em Python é um objeto!

■ Introdução a Linguagem de Programação Python

■ O que é o Python?

■ Princípios pythonicos:

- O que são PEP's? Python Enhancement Proposals - Propostas para Melhoramento no Python;
- São guias e procedimentos para melhor utilização da linguagem;
- Auxilia na produção de códigos mais limpos;
- PEP 20 (>>> import this)
 - Simples é melhor que complexo;
 - Legibilidade conta;
 - Casos especiais não são especiais o bastante para quebrar as regras;
- PEP 8 – Convenções de estilo: tornar o código mais legível e coerente;
 - <https://www.python.org/dev/peps/pep-0008/>
 - Auxilia a manter a consistência na produção de códigos;
 - Indentação: utilize 4 espaços;
 - Funções e variáveis: utilizar snake case; nome_aluno; def calcular_salario();
 - Classes: utilizar camel case; class ContaEspecial(Conta):

■ Introdução a Linguagem de Programação Python

■ O que é o Python?

- Linguagem simples, com curva de aprendizado inicial bem elevada;
- Abstração elevada: semelhante construir de algoritmos na língua inglesa;

```
Algoritmo
  Declare n Numérico
  Leia n
  Se resto (n, 2) = 0
    então Escreva 'É par',
    senão Escreva 'É ímpar'
  Fim se
Fim algoritmo
```

```
n = input('Favor insira um número inteiro: ')
n = int(n)
if n%2 == 0:
    print('O número n: {:d}, é par'.format(n))
else:
    print('O número n: {:d}, é ímpar'.format(n))
```

■ Introdução a Linguagem de Programação Python

- Uma das principais linguagens de programação utilizadas atualmente.
- Classificada como a 2ª linguagem mais procurada -> Stackoverflow Survey 2019
 - <https://insights.stackoverflow.com/survey/2019#technology-most-loved-dreaded-and-wanted-languages>
- Classificada como a 4ª linguagem mais popular -> Stackoverflow Survey 2019
 - <https://insights.stackoverflow.com/survey/2019#technology-programming-scripting-and-markup-languages>

■ Introdução a Linguagem de Programação Python

- Utilizada por diversas empresas de grande porte



■ Introdução a Linguagem de Programação Python

■ Instalações necessárias:

■ Python 3:

- <https://www.python.org/downloads/>

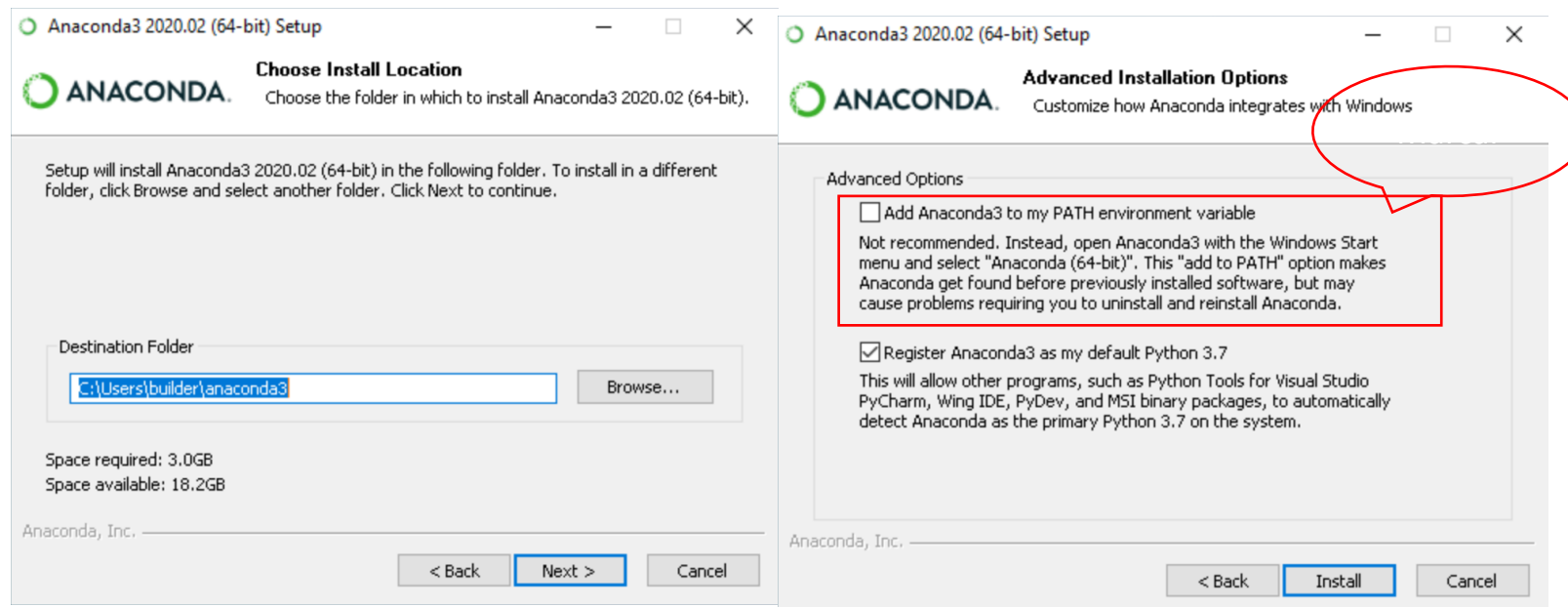
■ Sugestão: trabalhe com gerenciadores de ambientes de desenvolvimento

- Exemplo: Anaconda ou Virtual env
- Gerenciadores de ambiente de desenvolvimento permite que o programador possua diversas versões do interpretador e de bibliotecas instaladas no seu ambiente de desenvolvimento;
- Anaconda possui gerenciador de pacotes conda e pip;
- Permite a rápida replicação de ambientes, auxiliando no desenvolvimento em equipe e na construção de software;
- Fácil integração com diversos ambientes de desenvolvimento integrados:
 - VS Code, Pycharm, IDLE, etc.

■ Introdução a Linguagem de Programação Python

■ Instalando o Anaconda:

- <https://www.anaconda.com/products/individual#Downloads>



■ Testes:

- Abra o prompt e digite:
- `conda --version`
- `python -V`

■ Introdução a Linguagem de Programação Python

■ Instalações necessárias:

■ No Linux (baseado em Debian):

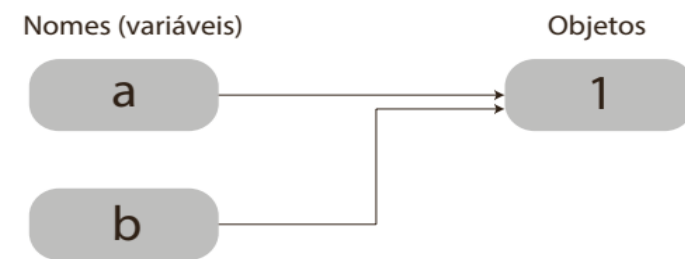
- Baixe a versão para seu sistema;
- `$ bash <O NOME DO ARQUIVO>.sh` ou `gedit <O NOME DO ARQUIVO>.sh`
- Quando perguntar se deseja adicionar o Anaconda no PATH, digite: `yes`
- Para Windows e UNIX: caso não deseje que seu ambiente seja ativado de início:
- `conda config --set auto_activate_base false`
- Teste: `conda activate base`
- `python --version`

■ Introdução a Linguagem de Programação Python

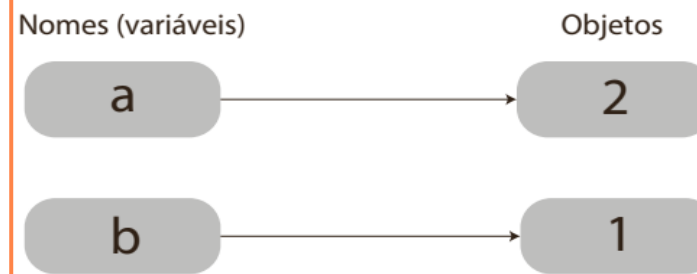
■ Variáveis e constantes:

- Referem-se a um espaço em memória através de um nome;
- Servem para armazenar valores;
- Devem ser iniciadas com uma letra ou sublinhado (_);
- São case sensitive; nomealuno ≠ nomeAluno;
- Observando os princípios Pythonicos, vamos utilizar o snake case: nome_aluno;
- Nomes de variáveis e funções devem representar sua ação -> nomes claros e significativos;
- Atribuição de valor através do sinal de igualdade (=);
- Sinal de igualdade em Python não tem significado matemático;
- Exemplo: nome_aluno = 'Maria Torres' // nota_1 = 9.50

a = 1 (a recebe o valor 1)
b = a (b recebe o valor da variável a)



a = 2 (a recebe o valor 2)
b não muda de valor com a mudança do valor de a



■ Introdução à linguagem Python

■ Palavras reservadas ou palavras-chave:

- Algumas palavras não podem ser utilizadas para declaração de variáveis;
- Irá gerar um erro de sintaxe

and	as	assert	break	class
continue	def	del	elif	else
except	exec	if	import	in
is	lambda	not	or	pass
print	raise	return	try	while
yield	True	False	None	

```
Prompt de Comando - python
Microsoft Windows [versão 10.0.19041.450]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\ricka>python
Python 3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license()" for more information.
>>> class = 3
      File "<stdin>", line 1
        class = 3
          ^
SyntaxError: invalid syntax
>>>
```



■ Introdução à linguagem Python

■ Os principais tipos de variáveis na linguagem Python são:

- bool: assume somente 02 valores: True ou False;
- int: representa números inteiros, por exemplo: `numero_n = 10`;
- float: representa números reais. Parte decimal utiliza-se “.” (ponto) e não “,” (virgula), por exemplo: `PI = 3.14`;
- str: representa uma sequência de caracteres ou texto;
- None: é um objeto que representa um espaço vazio, ou seja, naquele espaço de memória alocada para uma variável não possui nenhum valor;

■ Transformação de variáveis:

- Texto para números:
 - Todo input realizado no terminal é recebido como uma string;
 - `nota_1 = float(input("Favor insira a nota da P1: "))` -> `nota_1` -> tipo float
 - `nota_1 = input("Favor insira a nota da P1: ")` -> `nota_1` -> tipo str
 - `nota_1 = float(nota_1)` -> `nota_1` -> tipo float
 - Somente a parte inteira: `nota_1 = int(nota_1)` -> `nota_1` -> tipo int
- Números para números:
 - `nota_inteira = int(nota_1)` -> `nota_inteira` irá retornar o valor igual à 9 -> tipo int
 - `nota_float = float(7)` -> `nota_float` irá retornar o valor igual à 7.0 -> tipo float



■ Introdução à linguagem Python

■ Alguns métodos da classe string:

- `capitalize()`
 - Retorna a string com o primeiro caractere maiúsculo;
- `count()`
 - Retorna o número de ocorrências de uma substring;
- `lower()`
 - Retorna a string com todos os caracteres minúsculos;
- `split()`
 - Quebra uma string em substrings;
- `strip()`
 - Retorna a string com os espaços iniciais e finais removidos;
- `upper()`
 - Retorna a string com os caracteres todos em maiúsculo;

■ Introdução à linguagem Python

■ Comandos de entrada e saída:

input() -> Entrada de dados;

Ex.: `nota_1 = float(input("Favor insira a nota da P1: "))`

Receberá o valor digitado pelo usuário e fará a mudança para uma variável de número decimal (float).

print() -> Saída de informação;

Ex.: `print("A média do aluno: {0} foi igual à: {1}".format(nome_aluno, media_aluno))`

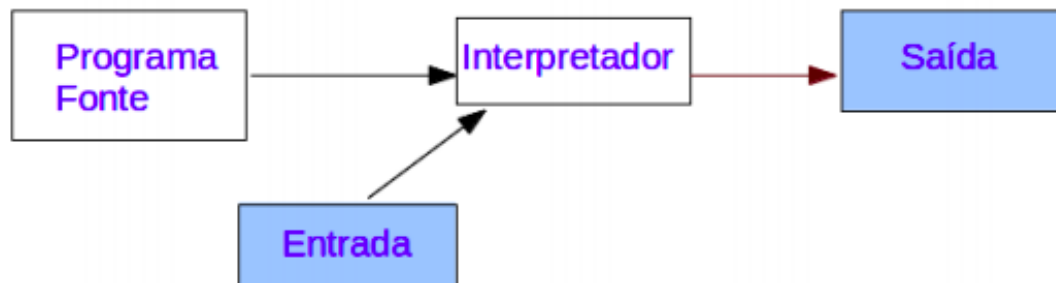


Figura 1: Funcionamento de um programa de computador com linguagem interpretada, como o Python.

■ Introdução à linguagem Python

- Escreva um código em Python para receber uma frase do usuário e transformar todos os caracteres em maiúsculo, minúsculo e capitalizado;
- Escreva um código em Python para receber dois valores e separar em duas variáveis;
- E em 3 variáveis?

■ Operadores Aritméticos

- Soma:



- Resto de divisão:



- Subtração:



- Parte inteira de uma divisão:



- Multiplicação:



- Divisão:



- Potenciação:



37 : 12 = 3 com resto 1


$$\begin{array}{r}
 \text{Dividendo} \rightarrow 37 \overline{) 12} \leftarrow \text{Divisor} \\
 \underline{-36} \quad 3 \leftarrow \text{Quociente} \\
 1 \leftarrow \text{Resto}
 \end{array}$$

■ Introdução à linguagem Python

■ Outros operadores aritméticos:

Operador	Exemplo	Descrição
<code>+=</code>	<code>a += b</code>	Mesmo que: <code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	Mesmo que: <code>a = a - b</code>
<code>*=</code>	<code>a *= b</code>	Mesmo que: <code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	Mesmo que: <code>a = a / b</code>
<code>%=</code>	<code>a %= b</code>	Mesmo que: <code>a = a % b</code>
<code>**=</code>	<code>a **= b</code>	Mesmo que <code>a = a ** b</code>

■ Operadores lógicos e relacionais

- São utilizados para realização de comparação  processo decisório!
 - Equivalência ou similaridade: igual ou diferente;
 - Qualidade: melhor ou pior;
 - Dimensão: maior ou menor;
 - Quantidade: mais ou menos, muito ou pouco;

■ Tipos de operadores relacionais:

- $A == B$ (A igual a B)
- $A != B$ (A diferente de B)
- $A > B$ (A maior do que B)
- $A >= B$ (A maior ou igual a B)
- $A < B$ (A menor do que B)
- $A <= B$ (A menor ou igual a B)
- $A \text{ is } B$ (Mesmo objeto)
- $A \text{ in } B$ (Está contido)

■ Tipos de operadores lógicos:

- $E \Rightarrow$ and ou $\&$ (duas expressões sejam verdadeiras);
- Ou \Rightarrow or ou $|$ (pelo menos uma expressão verdadeira);
- Não \Rightarrow not ou \sim (negação de uma expressão)



■ Introdução à linguagem Python

■ Regras de precedência:

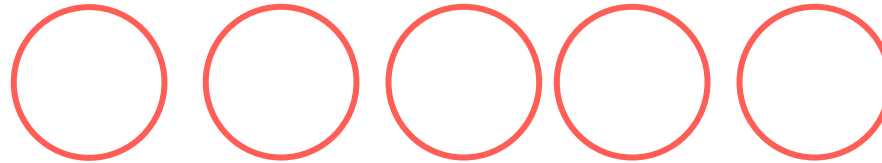
1. Expressões entre parênteses;
2. Exponenciação;
3. Multiplicação, divisão, resto e parte inteira;
4. Soma e subtração;
5. Operadores relacionais: $<$, $<=$, $>$, $>=$, $!=$, $==$
6. not
7. and
8. Or

OBS.: Esquerda para direita;

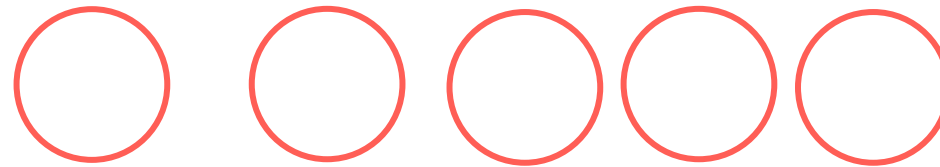
■ Introdução à linguagem Python

- Qual a sequência das operações abaixo:

$r = a * b \% c + d / e - f$



$r = a * b ** 2 + c * d + e$



■ Introdução à linguagem Python

- Qual a sequência das operações abaixo:

$r = a * b \% c + d / e - f$

1 2 4 3 5

$r = a * b ** 2 + c * d + e$

2 1 4 3 5

- **Introdução à linguagem Python**
- Implementar algoritmo “Calcular média de dois números inteiros” em Python

```
numero_1=10  
numero_2=20  
media=(numero_1+numero_2)/2  
print(media)
```

- **Tabela verdade**

```
proposicao_a = True  
proposicao_b = False  
resultado = proposicao_a and proposicao_b  
print(resultado)
```

■ Introdução à linguagem Python

- Elabore um código em Python para expressar a tabela verdade de “e” , “ou” , “negação”;
- Elabore um código em Python para calcular a média aritmética de 2 números inteiros;
- Elabore um código em Python para calcular a média ponderada de 2 números inteiros;
- Elabore um código em Python para receber do usuário:
 - Nome do aluno;
 - 03 notas;
 - Retorne uma mensagem para o usuário informando a média aritmética desse aluno;
- Elabore um código em Python para receber do usuário:
 - Nome do aluno;
 - 02 notas;
 - 02 pesos, respectivamente para cada nota;
 - Retorne uma mensagem para o usuário informando a média ponderada desse aluno;
 - Utilize o método split para separar a entrada do usuário em nota e peso. Exemplo: Digite a nota e peso: 9.5,3
- Elabore um script em Python para executar o cálculo da equação do segundo grau
 - Sem biblioteca math;
 - Com biblioteca math:
 - `import math;`
 - `>>> math.sqrt(9)`
 - `>>> 3`



■ Introdução a Linguagem de Programação Python

- Referências:

- PERKOVIC, Ljubomir. Introdução à Computação usando Python. Rio de Janeiro: LTC, 2016.
- <https://docs.python.org/3.8/tutorial/index.html>
- <https://www.python.org/dev/peps/>
- <https://www.w3schools.com/python/>



UNIVERSIDADE
CANDIDO
MENDES

EAD ■

