

AC-309 – Atividades Complementares

(Engenharias Biomédica e de Telecomunicações)



Prof. Evandro Luís Brandão Gomes

Plano de Ensino

OBJETIVOS GERAIS

Compreensão e resolução de problemas computacionais e de engenharia por meio da Linguagem de Programação *python*, revisão e aplicação de conceitos fundamentais da Orientação a Objetos nesta linguagem.

Plano de Ensino

CONTEÚDO PROGRAMÁTICO

- 1.Introdução a Linguagem python
- 2.Tipos de Dados, Variáveis e Operadores
- 3.Controle de Fluxo
- 4.Introdução a Orientação a Objetos com python
- 5.Coleções
- 6.Herança, Reescrita e Polimorfismo

Plano de Ensino

CRITÉRIOS E PROCEDIMENTOS DE AVALIAÇÃO

Para a validação das horas o aluno precisa ter **frequência mínima de 75%** nas atividades e **média maior ou igual a 50 pontos**.

A NF (Nota Final) será calculada da seguinte forma:

$$NF = (0,75*N) + (0,25*R)$$

Onde:

N valerá 100 pontos, sendo 80 pontos de uma prova teórico-prático e 20 pontos de exercícios.

R é a nota de redação na Atividade Complementar.

Plano de Ensino

CRITÉRIOS E PROCEDIMENTOS DE AVALIAÇÃO

Se $NF \geq 50$ o aluno está **aprovado com média NF**, desde que tenha **frequência mínima de 75%** nos encontros presenciais da atividade.

Se $NF < 50$ o aluno **deverá fazer a AG** e a **NF será recalculada** por:

$$NF = 0,75*(0,5*N + 0,5*AG) + (0,25*R)$$

Se $NF \geq 50$, então o aluno está aprovado, com média NF, desde que tenha frequência mínima de 75% nos encontros presenciais da atividade.

Se $NF < 50$ o aluno estará reprovado.

Capítulo 1

Fundamentos de *python*



O que é python?



Python é uma linguagem de programação relativamente **simples** que foi criada em 1991, ela é de **alto nível**, **interpretada** e de alta produtividade.

Simples

- *Menos linhas de código comparando com o Java, C, C++ (mais elegante)*
- Documentação Gratuita e de fácil acesso

Alto nível

- Longe do código de máquina
- Próximo à linguagem humana



O que é PYTHON?



Python é uma linguagem de programação relativamente **simples** que foi criada em 1991, ela é de **alto nível**, **interpretada** e de alta produtividade.

Interpretada

- O código fonte é executado por um programa tradutor, evita o “codifica-compila-executa”





Um pouco de História



- Em 1989, Guido Van Rossum, um programador Holandês, cansado das dificuldades e complexidades da linguagem C, resolveu criar uma nova linguagem de programação;
- A ideia dele era criar uma linguagem tão poderosa quanto C, porém de mais fácil compreensão e forma de programar;
- Colocou o nome de python em homenagem ao seu programa de TV favorito: “Monty Python Flying Circus”.



Guido van Rossum,
o criador do Python

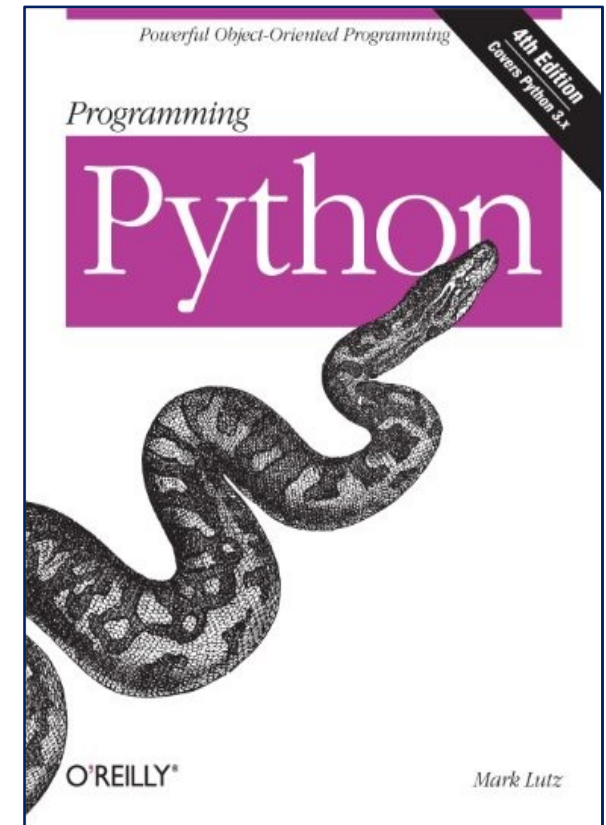




Um pouco de História



- O primeiro livro de python foi publicado pela editora O'Reilly, que tem como costume colocar animais nas capas de suas obras;
- Como python também é o nome de uma espécie de serpente, a editora resolveu colocar esse animal como capa do livro de python;
- Isso acabou dando origem ao seu logotipo das duas serpentes entrelaçadas.





Um pouco de História



- Pela simplicidade oferecida pela linguagem, a mesma teve seu “boom” na década de 90 e começou a se popularizar rapidamente entre os programadores;
- Com o fortalecimento da comunidade de desenvolvedores e financiamento de grandes empresas nos EUA, em 2001 foi criada a “**Python Software Foundation**”, uma fundação sem fins lucrativos que funciona como mantenedora da linguagem até os dias atuais.





Por que usar python?



Com tantas linguagens de programação, por que usar a linguagem python?



1. É uma linguagem de propósito geral. Pode-se usar python para criar qualquer tipo de programa;
2. É uma das linguagens mais fáceis e intuitivas (comandos muito simples);
3. É multiplataforma (o mesmo programa executa em diferentes sistemas operacionais e dispositivos);
4. Possui um conjunto de bibliotecas muito completa;
5. É um software livre (não é necessário pagar para usar ou distribuir);
6. É muito organizada (força o programador a deixar o código organizado);
7. É orientada a objetos (paradigma de programação mais popular atualmente);



Quem usa python?



Já vem até instalado de fábrica
nestes sistemas operacionais!





Instalando o python



Como utilizaremos o sistema operacional **Windows**, um dos poucos que não vem com o python instalado, precisamos verificar se o interpretador python já foi instalado no computador.

- Entre no **Prompt de Comando** e digite:

```
C:\>python --version
```

Se o python estiver instalado você verá a mensagem da versão instalada:

```
C:\>python --version
Python 3.8.1
C:\>
```

Se a mensagem for “**python não é reconhecido como um comando interno ou externo**”, será necessário fazer a instalação do python



Instalando o python



Fazer download no site oficial www.python.org, na opção de download e depois fazer a instalação (versão 64bits).

The screenshot shows the Python.org website. The 'Downloads' tab is selected. The 'Download for' dropdown menu is open, and the 'Windows' option is highlighted with a red box. A red arrow points from this box to the 'Download Windows x86-64 executable installer' link in the 'Stable Releases' section.

Python >>> Downloads >>> Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.8.1
- Latest Python 2 Release - Python 2.7.17

Stable Releases

- Python 3.8.1 - Dec. 18, 2019

Note that Python 3.8.1 cannot be used on Windows XP or earlier.

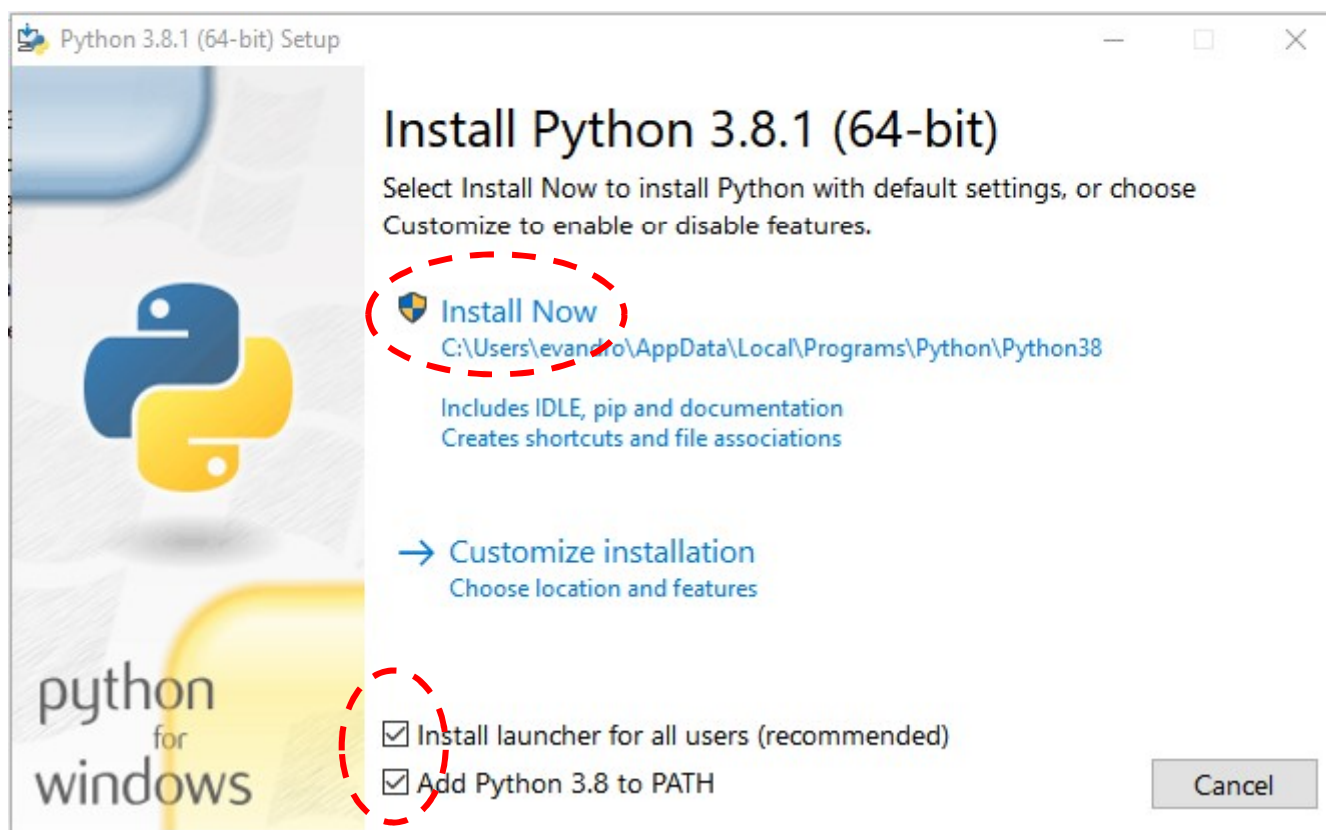
- Download [Windows help file](#)
- Download [Windows x86-64 embeddable zip file](#)
- Download [Windows x86-64 executable installer](#)
- Download [Windows x86-64 web-based installer](#)



Instalando o python



Na tela de instalação, não esqueça de marcar as duas opções disponíveis embaixo:





Python – Primeiros passos



Após a instalação, em “Executar” digite “python” ou entre no “*Prompt de Comando*” e digite “python”:

```
C:\>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Este é o modo console do interpretador python. Nele é possível digitar comando por comando e ver o resultado.

Tente alguns comandos



Python – Primeiros passos



No modo console é possível testar comandos do interpretador python:

Exemplos:

```
>>>10+5
15
>>>print("Inatel")
Inatel
>>>print('Aula de AC-309')
Aula de AC-309
>>>nome = "AC-309"
>>>valor = 10
>>>print(nome,valor)
AC-309 10
>>>
```



Python – Primeiros passos



Exercício:

Digite os comandos em python que armazene em variáveis o nome, a idade e o peso de uma pessoa depois mostre os valores.



Python – Primeiros passos



Como ler dados do teclado? Através do comando `input()`

```
>>>valor = input("Digite um número:")  
Digite um número: 56  
>>>print("O número digitado foi ", valor)  
O número digitado foi 56  
>>>
```



Python – Primeiros passos



Exercício:

Reescreva o exercício anterior, porém agora, lendo os dados do teclado e armazenando em variáveis, depois mostre os valores.



Python – Primeiros passos



E agora se quisermos repetir estes comandos para entrada de novos dados?

Solução: temos que reescrever tudo novamente

Ou

Podemos salvar estes comandos em um “script” python e submeter ao interpretador todo vez que for necessário.

```
C:\>python prog1.py
```



IDEs para Desenvolvimento



Existem diversas IDEs (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) para desenvolvimento de programas em python, entre elas:

- IDLE → É instalada junto com o interpretador python. Bastante simples e intuitiva.
- Spyder → Bastante poderosa, com edição avançada, testes interativos, recursos de depuração.
- PyCharm → Muito completo. Interface simples e personalizável, porém muito pesada.
- Visual Studio Code (VS Code) → interface simples, bastante leve. Multilinguagem.
- Entre outras: Anaconda, Atom, WingIDE, etc.

A melhor IDE é aquela que te faz bem!



IDEs para Desenvolvimento



Usando a IDLE (Integrated Development and Learning Environment) que acompanha o python em sua instalação.

É uma maneira rápida de executar códigos python sem precisar abrir o Prompt de Comando do Windows:

The screenshot shows two windows from the Python 3.8.1 environment. The top window, titled 'Python 3.8.1 Shell', displays the Python version and system information, followed by a prompt 'Type "help", "copyright", "credits" or "license()" for more information.' and a red '>>>' prompt. The bottom window, titled 'prog1.py - D:\Aulas-2020\INATEL\1oSemestre\AC-309\prog1.py (3.8.1)', shows a Python script with the following code:

```
nome=input("Entre com o nome:")
idade=input("Entre com a idade:")
peso=input("Entre com o peso:")
print("Nome:",nome, " Idade=",idade, " anos  peso=",peso, " Kg")
```




Exercícios:

Usando a IDLE faça os seguintes scripts em python:

- 1- Crie um script que leia o nome de uma pessoa e mostre uma mensagem de boas-vindas colocando o nome digitado.
- 2- Crie um script que leia dia, mês e ano de nascimento de uma pessoa e mostre uma mensagem com a data formatada.
- 3- Crie um script que leia dois números e mostre a soma deles. Observe atentamente o resultado ... Está correto? Por que?




Exercícios propostos



Acesse o site oficial de exercícios propostos para python em:

<https://wiki.python.org.br/ListaDeExercicios>

E faça os exercícios de
“EstruturaSequencial”



The screenshot shows the web page wiki.python.org.br/ListaDeExercicios. The page header includes a login status "Logado como (Entrar)" and links for "associação" and "pythonbr". The main content area features the "python™ brasil" logo and a search bar. The title "ListaDeExercicios" is prominently displayed. Below the title, a paragraph explains that the list contains suggestions for programs for beginners in programming, covering various languages like Python, C, C++, Java, and Pascal. It also mentions that exercises are categorized by difficulty. A numbered list of nine links is provided: 1. [EstruturaSequencial](#), 2. [EstruturaDeDecisao](#), 3. [EstruturaDeRepeticao](#), 4. [ExerciciosListas](#), 5. [ExerciciosFuncoes](#), 6. [ExerciciosComStrings](#), 7. [ExerciciosArquivos](#), 8. [ExerciciosClasses](#), and 9. [ListaDeExerciciosProjetos](#). At the bottom, there is a section for sending suggestions and a link to the Python documentation.

Esta é uma lista com sugestões de programas para iniciantes em programação. Os exercícios podem ser implementados em pseudo-código, Python, C, C++, Java, Pascal ou em qualquer outra linguagem. Os exercícios estão divididos em categorias e procuram obedecer uma ordem de dificuldade crescente. Cada exercício é independente do outro, sendo possível deixar exercícios para trás sem grandes prejuízos.

Lista de exercícios:

1. [EstruturaSequencial](#)
2. [EstruturaDeDecisao](#)
3. [EstruturaDeRepeticao](#)
4. [ExerciciosListas](#)
5. [ExerciciosFuncoes](#)
6. [ExerciciosComStrings](#)
7. [ExerciciosArquivos](#)
8. [ExerciciosClasses](#)
9. [ListaDeExerciciosProjetos](#)

Envie sugestões sobre novos enunciados e melhorias nos textos já existentes. A criação de telas de exemplo com a saída dos programas também é algo importante de ser feito.

Está com dúvidas? Acesse a [DocumentacaoPython](#) e o material [P3 - Princípios de Programação com Python](#).