ECM251 - Linguagens I

Retomada do Conceito de Orientação a Objetos
Pilares da Orientação a Objetos

Prof. Murilo Zanini de Carvalho

Antes de começar!

Clone seu repositório do Github

- Lembre-se sempre antes de iniciar uma aula, clonar seu repositório remoto e realizar as atividades nele.
- Para cada atividade desenvolvida, criar um novo diretório.

Retirado de (https://miro.medium.com/max/4000/0*MZMl76wKo2FQLqG0.png), em 07/03/2021

ATENÇÃO PARA AULA DE HOJE!!

Implemente os códigos da aula, isso vai ser importante!!

Retirado de (https://www.testbytes.net/wp-content/uploads/2019/06/Untitled-63.png), em 22/05/2021

Retomada do Conceito de Orientação a Objetos

Objetivos da aula de hoje

- Deixar clara a diferença entre a abordagem procedural e a abordagem orientada a objetos.
- Estudar os conceitos fundamentais do paradigma orientado a objetos.
- Reescrever programas com orientação procedural para orientação a objetos.

Um *food truck* e *um* restaurante, analogia de paradigmas

Um *food truck* e *um* restaurante, analogia de paradigmas

Em um food truck, todas as tarefas ficam por conta do funcionário que está trabalhando ali.

Ele é RESPONSÁVEL por fazer a comida, servir a comida, cobrar e pegar os pedidos com os clientes.

Essa abordagem funciona mas apresenta diversos problemas. Com o aumento do negócio, um número maior de clientes vai se acumulando e fica difícil atender a todos as RESPONSABILIDADES do negócio.

Um *food truck* e *um* restaurante, analogia de paradigmas

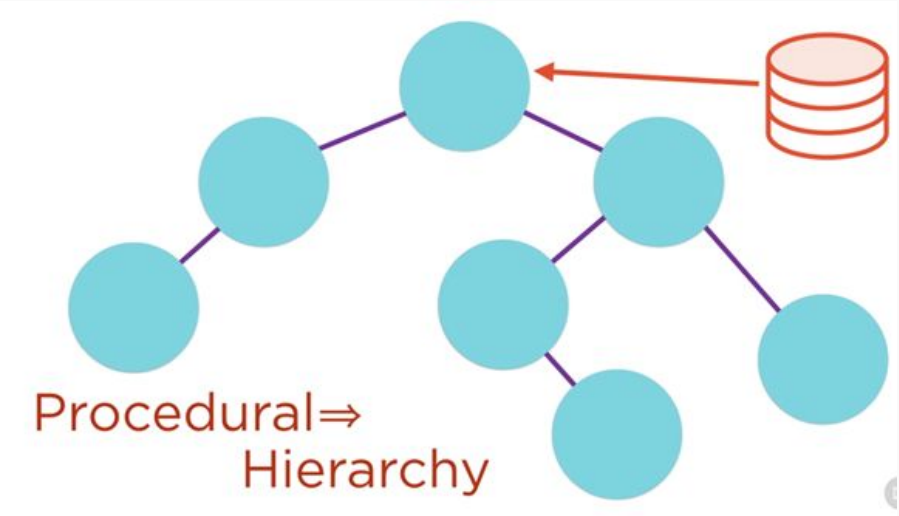
Já em um restaurante, existem diversos TRABALHOS. Cada funcionário fica com um conjunto de RESPONSABILIDADES.

Um garçom fica responsável por pegar os pedidos dos clientes e levar os pratos preparados para a mesa. Os funcionários da cozinha ficam responsáveis por preparar os pedidos que chegam e retornar os pratos com a comida. O caixa faz as cobranças.

Cada funcionário do restaurante realiza uma tarefa que, da sua INTERAÇÃO, faz com que o restaurante funcione.

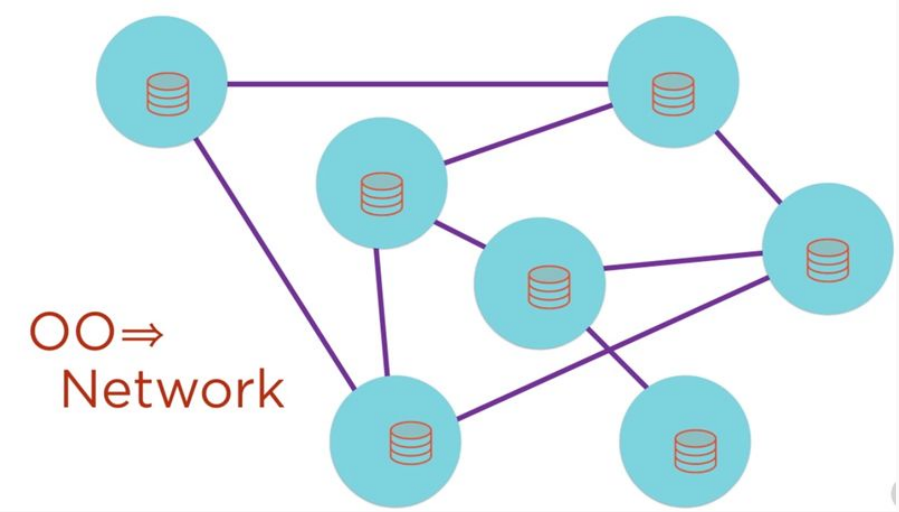
Na abordagem procedural

Data centric, e o fluxo de dados é muito importante. O sistema possui acesso a um conjunto de dados e todos os módulos alteram esses dados de alguma forma (Sanches, 2018).



Na abordagem orientada a objetos

OO não é data centric, parece um rede. Cada objeto será responsável por gerenciar o seu próprio conjunto de dados (Sanches, 2018).



Pilares da Orientação a Objetos

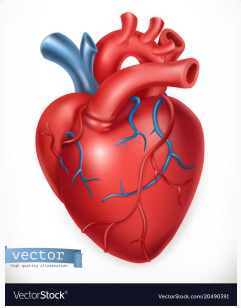
- Abstração
- Encapsulamento
- Herança
- Polimorfismo

Retirado de (https://www.vivaxsolutions.com/images/four-pillars.png), em 01/08/2019

Abstração

Os pontos fundamentais de um objeto ou problema são listados para representar esse objeto ou problema.

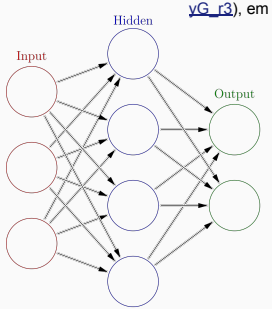
Quando essa representação fica muito complexa, ela deve ser dividida em abstrações mais simples. Cada abstração deve ter uma funcionalidade limitada, dessa forma, é possível gerenciar suas ações e seu impacto no projeto.



Retirado de (https://cdn2.vectorstock.com/i/1000x1000/03/91/human-heart-medicine-internal-organs-3d-icon-vector-20490391.jpg), em 01/08/2019

Retirado de (https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRSXKE4VaQV5UUxP6J-DiiQejB-arE9dhVxsKCNa1R3HsivG_r3), em 01/08/2019

Retirado de (https://cdn.the-scientist.com/assets/articleNo/36663/iImg/15248/d305ec2a-9f5a-4894-8cd3-a7c43bb0756b-brain-640.jpg), em 01/08/2019



Retirado de (https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Colored_neural_network.svg/1200px-Colored_neural_network.svg.png), em 01/08/2019

Encapsulamento

Quando um garçom vai fazer um pedido para o chef, ele não deve dizer ao chefe como ele deve cozinhar, ele apenas deve informar o que foi pedido. Cada classe é responsável por suas tarefas.Ainda assim, é necessário que o garçom possa fazer o pedido para o chef.

Para essa função, existe o local onde ele deve fazer esse pedido e receber de volta do chef o prato quando ele estiver pronto. Esse conceito é o da interface pública do objeto.

Retirado de (https://c8.alamy.com/comp/DX8DP7/chef-at-the-pass-plating-food-michelin-starred-the-three-chimneys-DX8DP7.jpg), em 01/08/2019

Herança

Quando as abstrações estão sendo construídas, podem existir métodos comuns em diversas classes propostas. Para retirar essa duplicação de código e para concentrar essas características comuns, é possível utilizar a herança.

Quando um objeto herda do outro, ele recebe seus métodos e atributos, tornando se também um objeto daquela classe.

Retirado de (https://s3.crackedcdn.com/phpimages/article/5/6/8/580568_v3.jpg), em 01/08/2019

Polimorfismo

Quando herdamos as características de um classe pai, alguns comportamentos podem precisar ser sobreescritos. Isso traz uma personalização para a classe filho.

Quando o método personalizado for chamado de uma instância da classe filha, ele vai ser chamado. Quando a chamada vier de uma instância da classe pai, o método original vai ser chamado.

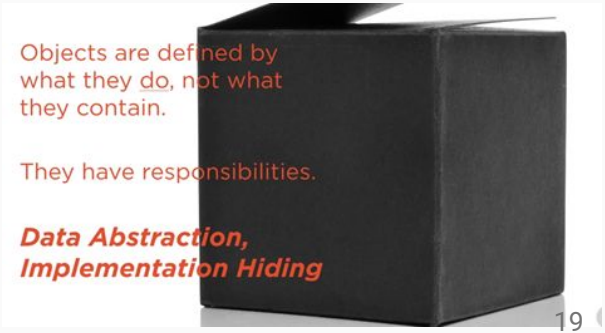
Retirado de (https://i.pinimg.com/originals/d4/6b/bc/d46bbc09f471eb370ccdf30aae8ae9a9.png), em 01/08/2019

Programas Orientados a Objeto

- Os objetos vão se conversar através de mensagens. Essas mensagens podem ser implementadas através de métodos, mas não é obrigatório.
- Métodos são funções que manuseiam as mensagens entrantes.
- Alguns métodos podem utilizar funções para fazer trabalhos simples de decodificação, cálculo ou algo do tipo.

Objetos

- Objetos são definidos pelo o que eles fazem, não pelo o que eles contêm. Eles devem ser vistos como uma caixa preta, que você pede para eles fazem algumas operações, e você não sabe qual a implementação e qual o processamento que ele irá realizar. Então podemos dizer que os objetos possuem responsabilidades, e devem realizar operações com coesão.
- Então não sabemos como o objeto funciona, mas sabemos como pedir coisas para eles, bem como o que esperar como resposta. O que existe dentro do objeto deve ser desconhecido para nós, como uma caixa preta.



Objetos - Princípio de Responsabilidade Única

Cada objeto deve possuir apenas uma responsabilidade. As duas imagens abaixo violam esse princípio.

Objetos - Acoplamento

Se você está pedindo informações para uma classe, isso pode ser indício que o acoplamento está sendo maior do que deveria.

Se precisamos realizar algum processamento com informações que o objeto possui, devemos então pedir para o objeto realizar o processamento e não pegar a informação para tal.

Objetos - Isolamento

Devo ser capaz de modificar qualquer variável, tipo ou implementação da minha classe, sem que as que utilizam ela sequer notam a diferença.

Cronograma

Cronograma

3 Bimestre:

- Aula 01 : Teoria - 10/08
  - Revisão dos pilares de POO
  - Revisão dos conceitos básicos de Python
- Aula 02 : Teoria - 17/08
  - Herança com Python
  - Polimorfismo com Python
- Aula 03 : Teoria - 31/08
  - Introdução a GUI com o usuário
- Aula 04 : Teoria - 24/08
  - Manipulação de arquivos com Python
  - Tratamento de exceções com Python
- Aula 05 : Teoria - 14/09
  - Dúvidas Gerais
- Aula 01 : Prática - 11/08
  - Introdução a POO com Python
- Aula 02 : Prática - 18/08
  - Exercícios com herança
  - Exercícios com polimorfismo
- Aula 03 : Prática - 01/09
  - Prática de GUI com o usuário
- Aula 04 : Prática - 25/08
  - Exercícios com Manipulação de arquivos com Python
  - Exercícios com tratamento de exceções
- Aula 05 : Prática - 08/09
  - Exercícios gerais
  - Exercícios com SQLite
- Aula 06 : Prática - 15/09
  - Atividade T3

Cronograma

4 Bimestre:

- Aula 01 : Teoria - 28/09
  - Introdução a banco de dados
- Aula 02 : Teoria - 05/10
  - Modelagem de dados relacional
  - Banco de dados SGBD
  - Postgres
- Aula 03 : Teoria - 19/10
  - Definição ORM
  - Implementando ORM
- Aula 04 : Teoria - 26/10
  - Padrões de Projeto com Python
- Aula 05 : Teoria - 09/11
  - Revisão Geral
- Aula 06 : Teoria - 16/11
  - Apresentação de resultados

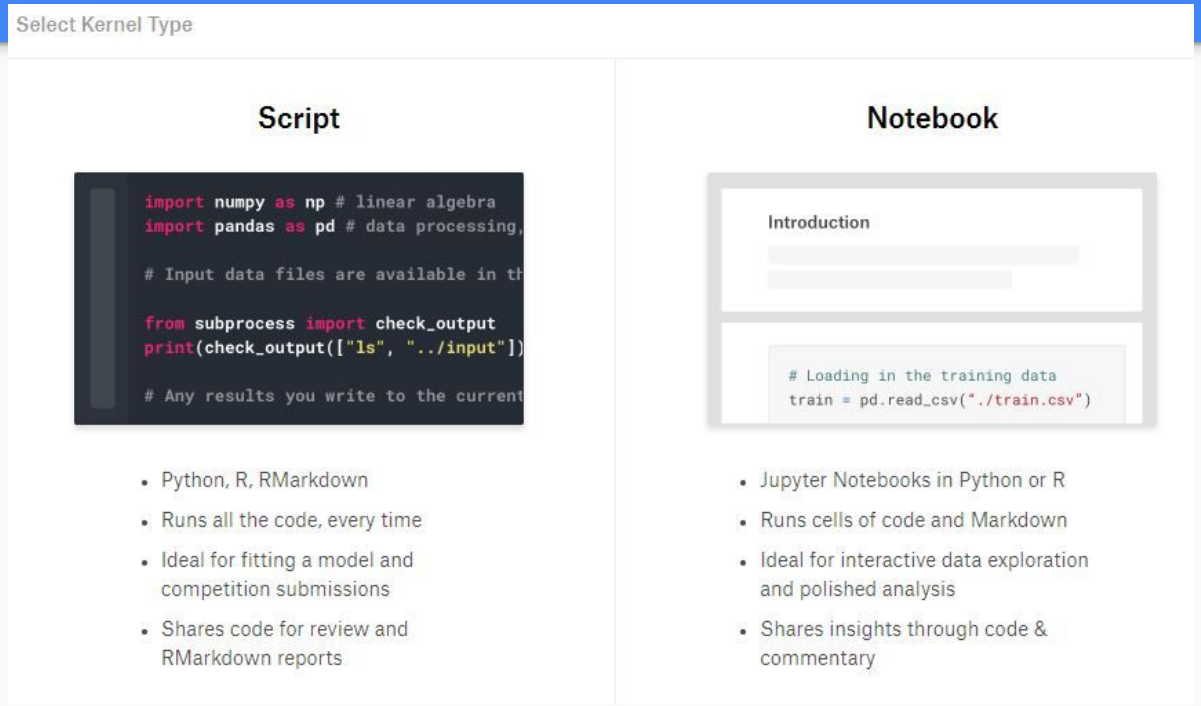
- Aula 01 : Prática - 29/09
  - SQLite
  - Banco de dados relacionais com Python
- Aula 02-A : Prática - 06/10
  - Acessando o Postgres com Python - Parte 1
- Aula 02-B : Prática - 13/10
  - Acessando o Postgres com Python - Parte 2
- Aula 03 : Prática - 20/10
  - SQL Alchemy
- Aula 04-A : Prática - 27/10
  - Implementando padrões de projeto com Python - Parte 1
- Aula 04-B : Prática - 03/11
  - Implementando padrões de projeto com Python - Parte 2
- Aula 05 : Prática - 10/11
  - Atividade T4

Revisão dos conceitos básicos de Python

Revisão dos conceitos básicos de Python

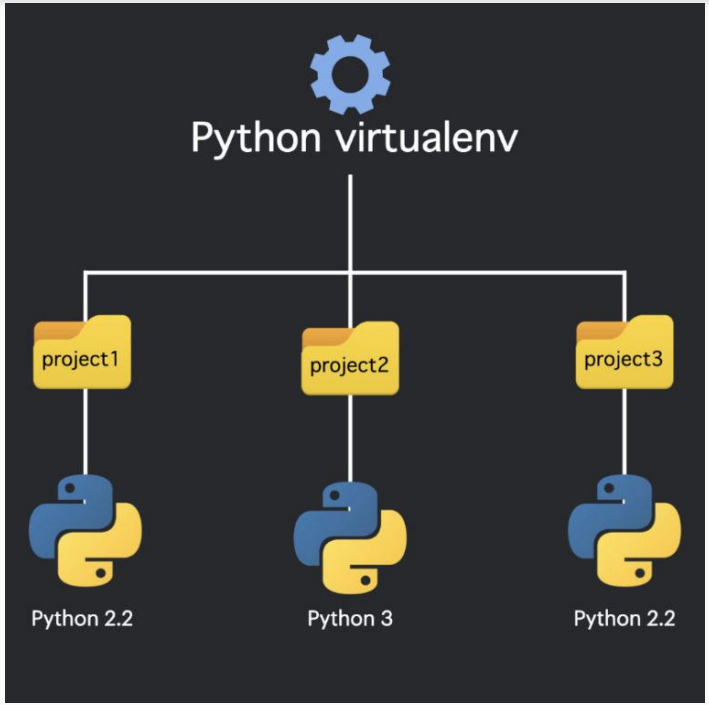
- Linguagem Multiparadigma (suporte ao paradigma Orientado a Objetos).
- Diversas versões no mercado.
- Possibilidade de utilização em diversas aplicações:
  - Backend
  - IA
  - Ciência de Dados

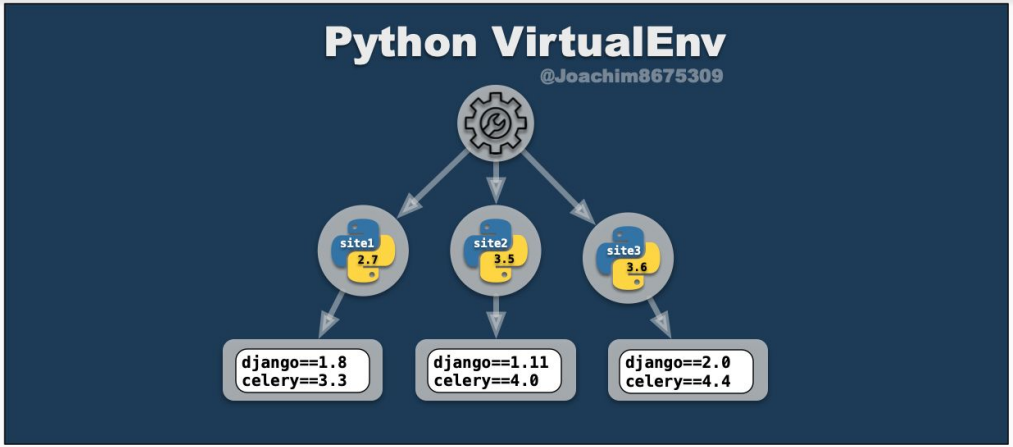
Revisão dos conceitos básicos de Python



Retirado de (https://1.bp.blogspot.com/-srqLioiiMPY/Wm6_08uBo4I/AAAAAAAAEZY/uLNCMW0xa18sVvR0ArykpToA79yNQ958QCLcBGAs/s1600/Capture.JPG), em 09/08/2022

Revisão dos conceitos básicos de Python

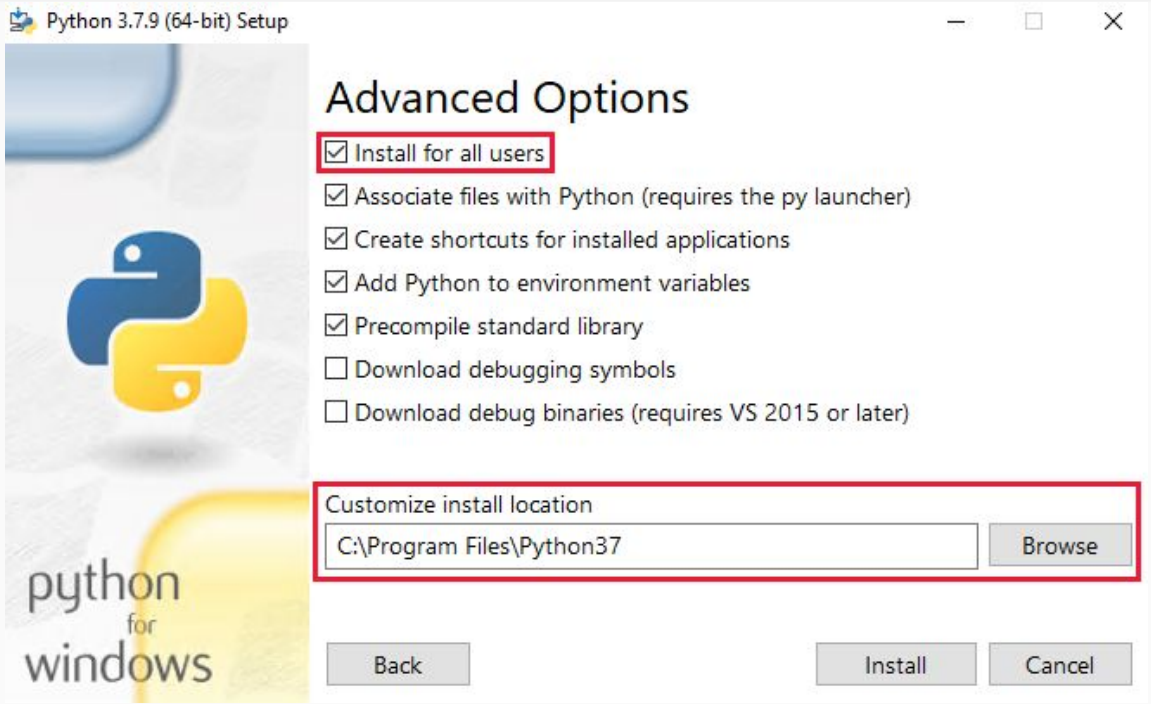




Retirado de (https://blog.debugeverything.com/wp-content/uploads/2021/04/python-virtualenv-project-structure.jpg), em 09/08/2022

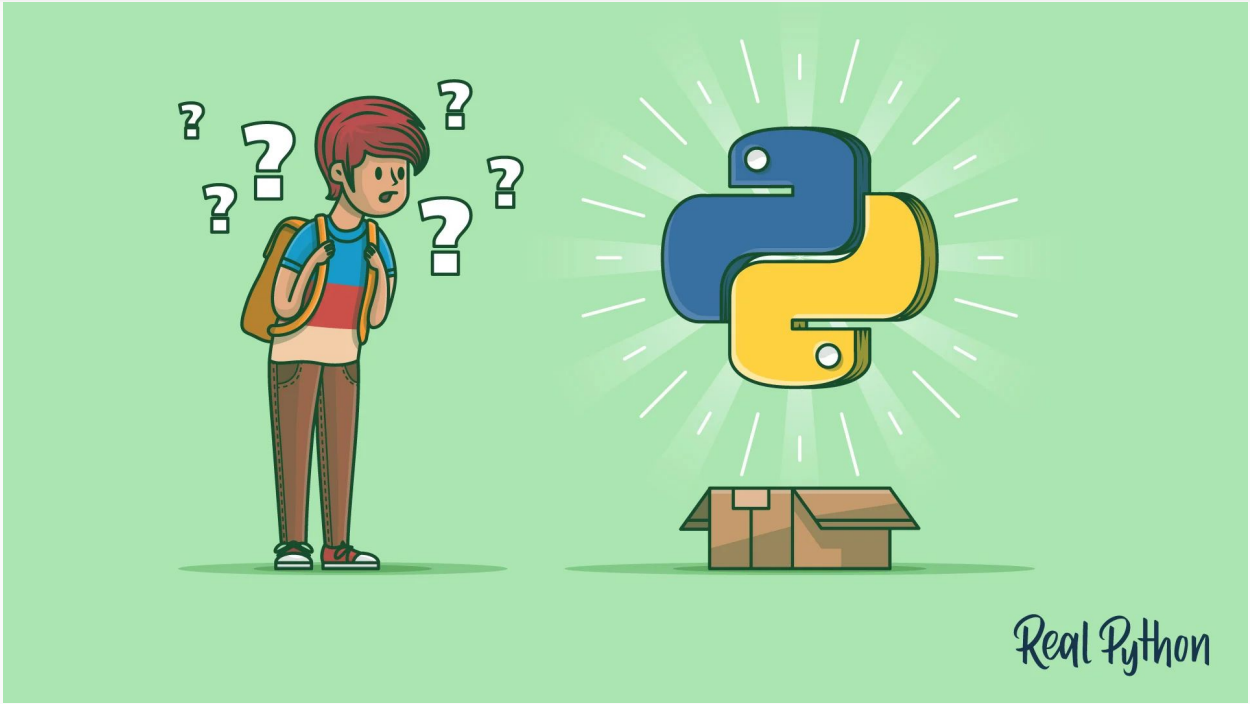
Retirado de (https://miro.medium.com/max/1838/1*wC-mrXuYgarKP8bSK3AivQ.png), em 09/08/2022

Revisão dos conceitos básicos de Python



Retirado de (https://docs.microsoft.com/pt-br/sql/machine-learning/install/media/python-install-for-all-users.png?view=sql-server-ver16), em 09/08/2022

Revisão dos conceitos básicos de Python



Retirado de (https://files.realpython.com/media/Newbie_Watermarked.a9319218252a.jpg), em 09/08/2022

Revisão dos conceitos básicos de Python

Retirado de (https://miro.medium.com/max/1134/1*OArR0xhMwgBsixQgldUIAw.png), em 09/08/2022

Vamos CODAR?

Retirado de (https://memegenerator.net/img/instances/65111891.jpg), em 09/08/2022

Revisão dos conceitos básicos de Python

- https://penseallen.github.io/PensePython2e/
- https://openbookproject.net/thinkcs/python/english3e/
- https://open.umn.edu/opentextbooks/textbooks/python-for-everybody-exploring-data-using-python-3

Perguntas?

Retirado de (https://cdn-icons-png.flaticon.com/512/1268/1268705.png), em 02/03/2022