

Linguagem Python II

Orientação a Objetos com Python

O que é do Orientação a Objetos? [↗](#)

De maneira introdutória, orientação a objetos consiste em programação orientada a objetos e como o próprio nome já expõem são linguagens e/ou técnicas de programação que possibilitam a utilização objetos. Breve e **resumidamente** serão introduzidos alguns dos diversos conceitos de OO:



Polimorfismo:

Se dá quando em uma hierarquia de classes existe a implementação métodos com a mesma assinatura, ou seja, o código funciona para qualquer classe desta hierarquia removendo a necessidade de implementações específicas para cada uma das classes.

Herança:

Existe quando uma classe herda os mesmos atributos e mesmos métodos, por exemplo, Cachorro e Gato são objetos da classe de mamíferos.

Encapsulamento:

O Encapsulamento se baseia no conceito de “esconder” os atributos de uma classe, ou seja, para quem for usar a classe não a utilizar de forma incorreta.

Benefícios da OO [↗](#)

- Diminui a complexidade para entendimento e manutenibilidade
- Orientação a Objetos facilita a reusabilidade de códigos, evitando a duplicidade.
- Facilita e acelera o desenvolvimento paralelo de classes e programas.
- Programas Orientados a Objetos são mais fáceis de testar, gerenciar e “

debuggar”.

Desvantagens da OO

- Possível superficialidade entre classes.
- Planejamento prévio e conhecimentos sobre programação são necessários.
- Muitas vezes classes acabam sendo generalizadas.

Curso Indicado

- Este curso irá introduzir conceitos que irão auxiliar na resolução dos exercícios adiante: [Python 3: introdução a Orientação a objetos](#)

Exercícios:

 [Fundamentos de Python - Lista I - Módulo II](#)

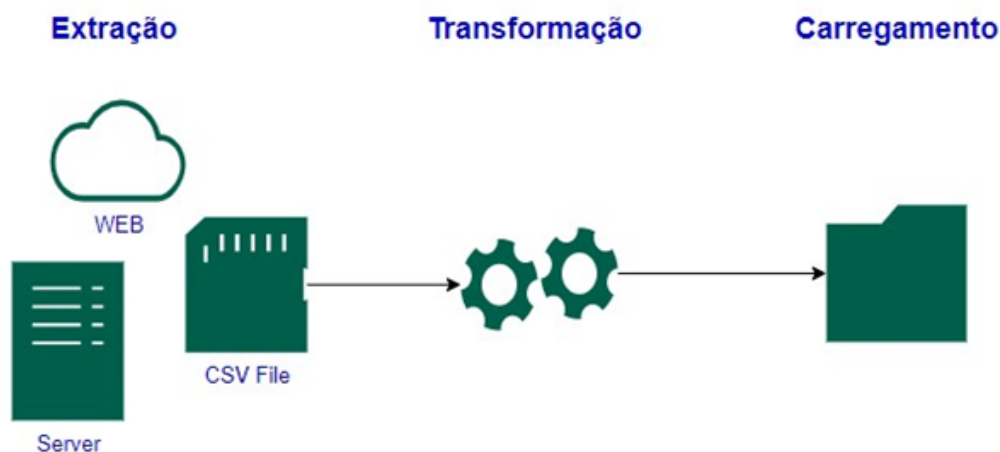
Extração, Transformação, Carregamento (ETL)

Como pode ser percebido pela primeira letra da sigla ETL, o primeiro processo a ser conduzido é a extração dos dados, ou seja, buscar os dados que devem ser armazenados na *staging area* (ou *landing zone*) de maneira que eles possam passar pelas transformações necessárias.

Dentre essas manipulações, diversas são usuais dentro do cenário da área de dados como por exemplo remoção de duplicatas, combinação de dados de fontes diversas para formação de novos *datasets* (*Conjuntos de dados*) e correção erros ortográficos por exemplo. Todas essas manipulações de dados citadas acima fazem parte dos processos de transformações que são os passos anteriores ao *Load* do ETL.

No último passo deste processo, ocorre o *Loading* dos dados aferidos. Neste estágio é de extrema importância assegurar a qualidade e integridade dos dados a serem publicados, de maneira que o usuário receba essa nova gama de informação com o resultado desejado.

O processo de ETL possui teorias mais extensas e essa introdução foi apenas uma forma de iniciar o assunto, no decorrer do programa de bolsas, os demais conceitos e exemplos serão apresentados com cursos e exercícios.



Exercícios:

 Fundamentos de Python - Lista II - Módulo II