Utilidade do Python

Python é uma linguagem de programação multifuncional (GPL) que roda basicamente em qualquer arquitetura de sistema e pode ser usada para uma grande variedade de aplicações em diferentes áreas, desde desenvolvimento web até aprendizado de máquina.

Além de ser versátil, o Python também é amigável para iniciantes. Por esses e outros fatores, é uma das linguagens de programação mais populares do mundo.

Se você quer saber mais sobre Python e seus usos mais comuns, você encontrou o artigo certo. Vamos explicar por que você deve aprender o básico da linguagem, para que ela serve e dar dez exemplos de grandes empresas que já a utilizam em suas operações.

Python na Web

Desenvolvimento Web inclui todas as atividades usadas para criar websites e aplicativos web-based. Existem duas partes em um Website – Client-side que no qual o código roda no browser do computador do usuário; e a Server-side, onde o código roda no servidor da web.

Por exemplo, o JavaScript é uma linguagem Client-side que roda no browser, e permite a interação do usuário em um website. Já o Python roda no servidor, e é responsável por processar a lógica dos Inputs dos usuários, interagindo com bases de dados e outros servidores, etc.

Desenvolvedores Python podem usar frameworks da própria Web para rapidamente e eficientemente construir aplicações dinâmicas da web, sem precisar ter que aprender uma linguagem client-side como o JavaScript. Frameworks reduzem significantemente o tempo de desenvolvimento através de automatização de tarefas da web comuns. A facilidade de uso e popularidade do Python para Desenvolvimento Web é umas das razões que empresas como Ideamotive se especializa em contratar e conectar empresas com talentosos web developers de Python.

O Python tem dois principais frameworks server-side – Django e Flask (que no qual é um micro-framework), Python também tem módulos nele mesmo que no qual ajudam a lidar com JSON, sockets, http-requests e mais, tornando o processo de web development mais fácil.

O que é o MySQL?

Trata-se de um **código aberto desenvolvido para a gestão de banco de dados**.

A origem de seu nome vem da junção de “My”, que é o nome da filha do cofundador, Michael Widenius, e “SQL”, que é a sigla para Structured Query Language, cuja tradução mais próxima é linguagem de consulta estruturada.

O MySQL é capaz de armazenar qualquer tipo de informação, desde um site institucional até um [e-commerce](https://rockcontent.com/br/blog/e-commerce-guia/). Isso inclui desde o registro de um simples dado até todo o inventário de produtos.

Finalidade do MySQL

O MySQL cria um banco de dados para armazenamento e manipulação de dados, definindo a relação de cada tabela. Clientes podem fazer solicitações digitando comandos SQL específicos no MySQL. A aplicação do servidor responde com a informação solicitada fazendo aparecer no cliente.

MySQL Connector

O **Mysql connector** permite a conexão das linguagens php, java, asp ou pyton com o **MySQL**. Assim, sua aplicação desktop, mobile ou web pode gerenciar as informações do banco de dados.

Como Usar?

utilizando o sistema de gerenciador de pacotes pip. Para isso, abra o terminal do seu sistema operacional e digite pip install **mysql**-**connector**-python ou para evitar possíveis erros, insira no final a versão do **mysql** instalado em sua máquina **usando** esse comando: pip install **mysql**-**connector**-python==8.0.13

JDBC Connector(Java Database Connectivity)

O JDBC é uma [API](https://rockcontent.com/br/blog/api/) — Application Programming Interface — do [Java](https://rockcontent.com/br/blog/o-que-e-java/) que contém uma série de classes e interfaces para realizar a comunicação entre uma aplicação desenvolvida em Java e o banco de dados utilizado por ela. É um recurso antigo do Java, pois está disponível desde a versão 1.1 do JDK — Java Development Kit —, que foi lançada em 1997.

Atualmente, faz parte da plataforma Java SE — Standard Edition — e está na versão 4.3. A arquitetura básica do JDBC é composta por duas camadas específicas:

* API JDBC, que é formada por um conjunto de classes e interfaces disponíveis nos pacotes java.sql e javax.sql, e é responsável pela comunicação entre a aplicação e o driver JDBC;
* API JDBC Driver: é um software que implementa os métodos necessários para realizar a comunicação com o banco de dados.

Existem quatro tipos de drivers JDBC. São eles:

* driver de ponte JDBC-ODBC: utiliza um driver ODBC para se conectar ao banco de dados;
* driver de API nativa: usa bibliotecas do lado do cliente para se comunicar com o banco de dados;
* driver de protocolo de rede: adota uma camada adicional de software para converter as chamadas JDBC no protocolo utilizado pelo banco de dados;
* driver fino ou Thin Driver: converte as chamadas JDBC de acordo com o banco de dados do fabricante.

Na prática

Na prática, o JDBC funciona como uma ponte, ou seja, **é uma camada intermediária entre a aplicação e o banco de dados.**Ele realiza a interpretação e a conversão dos comandos necessários para a manipulação do banco de dados, de acordo com as características de cada um.

Isso porque, apesar de todos os bancos de dados relacionais utilizarem a linguagem SQL — Structured Query Language —, existem diferenças entre as diversas bases existentes no mercado. Portanto, o JDBC é uma ferramenta responsável por realizar essa comunicação, conforme o banco acessado pela aplicação.

É importante dizer que é preciso utilizar o driver JDBC de acordo com a base de dados utilizada, ou seja, se formos utilizar um banco MySQL, devemos usar o JDBC apropriado para ele. Por isso, **cada empresa desenvolvedora de banco de dados disponibiliza o seu próprio driver para ser utilizado no Java**.

Ao utilizar o JDBC, devemos passar os dados necessários para fazer a conexão com o banco e as instruções SQL para realizar as operações na base, entre elas, inclusão, alteração, pesquisa ou exclusão de registros

Flask o que é?

Lançado em 2010 e desenvolvido por Armin Ronacher, o Flask é um micro-framework destinado principalmente a pequenas aplicações com requisitos mais simples, como por exemplo, a criação de um site básico.

Possui um núcleo simples e expansível que permite que um projeto possua apenas os recursos necessários para sua execução (conforme surja a necessidade, um novo pacote pode ser adicionado para incrementar as funcionalidades da aplicação).

Características do Flask

* Simplicidade: Por possuir apenas o necessário para o desenvolvimento de uma aplicação, um projeto escrito com Flask é mais simples se comparado aos frameworks maiores, já que a quantidade de arquivos é muito menor e sua arquitetura é muito mais simples.
* Rapidez no desenvolvimento: Com o Flask, o desenvolvedor se preocupa em apenas desenvolver o necessário para um projeto, sem a necessidade de realizar configurações que muitas vezes não são utilizadas.
* Projetos menores: Por possuir uma arquitetura muito simples (um único arquivo inicial) os projetos escritos em Flask tendem a ser menores e mais leves se comparados a frameworks maiores.
* Aplicações robustas: Apesar de ser um micro-framework, o Flask permite a criação de aplicações robustas, já que é totalmente personalizável, permitindo, caso necessário, a criação de uma arquitetura mais definida.

O que da pra fazer com Flask

O Flask fornece uma função auxiliar render\_template() que permite o uso do mecanismo modelo do Jinja. Isso facilitará bastante o gerenciamento do HTML ao escrever seu código HTML em arquivos html e ao usar lógica em seu código HTML.

Flask permite um rápido desenvolvimento de pequenas aplicações quando comparado a outros frameworks. Isso se deve por conta de que a ferramenta possui apenas os arquivos necessários para o desenvolvimento dessas aplicações

O que é um Blueprint Flask?

No Flask os Blueprints fazem exatamente esse mesmo papel, cada pedaço do seu projeto fica em um blueprint separado, ou seja, é como se você tivesse vários projetos em um só.

Definição geral de Blueprint

O blueprint de serviços é uma ferramenta que permite à empresa visualizar, em forma de diagrama, as relações entre diferentes componentes de um serviço (pessoas, processos e evidências, digitais ou físicas), que estejam diretamente conectadas a pontos de contato ao longo da jornada do consumidor.