Python

-Relatório-

A história do Python começa em dezembro de 1989, quando Guido van Rossum, um programador holandês, iniciou o desenvolvimento da linguagem como um projeto de hobby. Ele estava procurando criar uma linguagem que fosse fácil de entender e que incentivasse uma sintaxe clara e legível. A primeira versão do Python, 0.9.0, foi lançada em fevereiro de 1991, e já incluía algumas das características fundamentais que definiriam a linguagem, como exceções, funções e tipos de dados modulares.

-IDEs-

IDE é um pacote de software que consolida as ferramentas básicas necessárias para escrever e testar softwares.

* *Jupyter* fácil de usar, apresenta uma interface intuitiva e ideal para trabalhar com grandes conjuntos de dados para análise*;*
* *VS Code* é a IDE mais popular do mundo e não apenas para Python;
* *Pycharm IDE* dedicada ao Python;
* *Atom* editor de código aberto muito popular, tem uma interface bem amigável e conta com suporte integrado ao GIT e está disponível para os principais sistemas operacionais;
* *Wing Python IDE* extremamente leve e possui muitos recursos para turbinar a interpretação e debugging dos códigos;
* *Thonny* é um IDE python para iniciantes desenvolvido pela Universidade de Tartu na Estônia, foi projetado para tornar a vida mais fácil para novatos em Python;
* *IDLE Python IDE* lançado em dezembro de 1998 por Guido Van Rossum para o desenvolvimento Python*;*
* *Pydev;*
* *Elpy* não é para iniciantes!
* *Spyder Python IDE,*

-Aplicação-

Aplicado bastante em data science, machine learning, desenvolvimento de web, desenvolvimento de aplicativos, automação de scripts, fintechs, desenvolvimento de jogos, análise de dados e testagem de software.

-Bibliotecas-

Bibliotecas são uma coleção de subprogramas usados no desenvolvimento de softwares. A sua principal função é facilitar a programação, garantindo mais agilidade e menos erros.

* Para utilizar uma biblioteca em Python, você precisa primeiro instalá-la em seu ambiente de desenvolvimento. A maneira mais comum de fazer isso é usando um gerenciador de pacotes, como o pip. Após instalar a biblioteca desejada, você pode importá-la em seu código Python usando a declaração import.

-Comparação-

Comparado a outras linguagens, o Python se destaca por sua sintaxe legível, velocidade de execução, tipagem dinâmica e o ecossistema de bibliotecas e frameworks disponíveis. O Python oferece várias vantagens em relação a outras linguagens de programação. Sua facilidade de aprendizado, versatilidade, comunidade ativa, bibliotecas poderosas e recursos disponíveis são alguns dos principais pontos fortes do Python. Embora o desempenho possa ser uma preocupação em determinados casos, o Python compensa isso com seu rico ecossistema de desenvolvimento e ampla adoção em diferentes áreas.

---------------------------Questões------------------------------

### 1-O que é Python?

R: Python é uma linguagem de programação portátil de alto nível que possui gerenciamento automático de memória, threads, strings e é baseada em objetos. Uma linguagem de programação amplamente usada em aplicações da Web, desenvolvimento de software, ciência de dados e machine learning (ML).

2- Por que o Python é melhor que o Java?

R: o Python (quando comparado com o Java) é mais fácil de usar e possui velocidades de codificação muito melhores. Além disso, quando se trata de dados, o Java é estaticamente digitado, enquanto o Python oferece digitação dinâmica. Isso é considerado um grande avanço.

3- Quantos tipos de dados existem no Python?

R: O Python possui cinco tipos de dados diferentes: string, lista, número, dicionário e tupla.

4: Qual é a diferença entre uma 'tupla' e uma 'lista'?

R: A principal diferença é que as listas são mais lentas, mas podem ser editadas, enquanto as tuplas funcionam mais rápido, mas não podem ser modificadas.

5- O que é 'decapagem' e 'desinteressante'?

R: A decapagem acontece quando um módulo dentro do Python é aceito e convertido em um módulo de string, e depois é despejado no arquivo. Ao contrário disso, o desengatamento é quando você recupera o módulo de string do arquivo.

6- O que é 'lambda'?

R: Lambda é uma função anonimamente executada com apenas uma expressão única.

7- Como a memória é gerenciada no Python?

R: O espaço de heap privado do Python é responsável pelo gerenciamento de memória. É acessível apenas por um *intérprete* - se você for um programador Python, não poderá alcançá-lo. A linguagem também possui uma recicladora embutida que é responsável por criar mais espaço de pilha livre (isso é feito através da reciclagem de memória não usada).

8- O que é 'passar'?

R: Passe simplesmente indica um espaço que deve ser deixado em branco dentro da declaração composta.

9- Você pode copiar um objeto em Python?

R: Sim, você pode copiar objetos no Python, mas não todos eles. A maneira mais geral e conhecida de fazer isso é usar o comando **copy.copy ()**.

10- Como deletar um arquivo dentro do Python?

R: Para deletar algo no Python, use o comando **os. Remove (name\_of\_the\_file).**

11- O que é um 'dicionário'?

R: Os dicionários são compostos de chaves e os valores correspondentes da chave.

12- Python é uma linguagem interpretada?

R: Sim, o Python é uma linguagem de programação interpretada. O que isto significa? É um processo de três vias - você escreve o código-fonte, o Python o converte em uma linguagem intermediária (para facilitar o entendimento) e depois é novamente transformado em códigos de máquina que são então executados.

13: Qual destes está errado?

R: Anulada.

14: Como o Python é orientado a objeto?

R: Linguagens de programação orientada a objetos (OOPs) são baseadas em classes e objetos dessas classes. Python é exatamente isso. Mais ainda, o Python possui os recursos que são creditados aos OOPs - herança, polimorfismo, etc.

15: O que é 'fatiar'?

R: No Python, o fatiamento é quando você seleciona vários itens de lugares como listas, sequências de caracteres e assim por diante.