**RELATÓRIO**

A história do Python começa em dezembro de 1989, criadoGuido 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.

- IDES é onde codifica e roda o código. As principais IDES do Python são: PyCharm, IDLE, VS Code, Spyder e Atom.

- A linguagem Python tem vários casos de uso no desenvolvimento de aplicações incluindo o: Desenvolvimento Web do lado do servidor, Automação com scripts Python, Ciência de dados e machine learning, Desenvolvimento de software e Automação de testes de software.

- Uma biblioteca é uma coleção de códigos usados com frequência que os desenvolvedores podem incluir em seus programas Python para evitar escrever o código do zero. Por padrão, o Python vem com a Standard Library, que contém muitas funções reutilizáveis. Além disso, mais de 137 mil bibliotecas Python estão disponíveis para várias aplicações, incluindo desenvolvimento Web, ciência de dados e machine learning (ML). As mais populares são: OpenCV-Python, NumPy, Pandas, Requests etc.

- Um framework Python é uma coleção de pacotes e módulos. Um módulo é um conjunto de códigos relacionados, e um pacote é um conjunto de módulos. Os desenvolvedores podem usar frameworks Python para criar aplicações Python mais rapidamente, porque não precisam se preocupar com detalhes mínimos, a forma como as comunicações acontecem na aplicação Web nem como o Python tornará o programa mais rápido. O Python tem dois tipos de framework:

* **Um framework full stack**: inclui quase tudo o que é necessário para criar uma aplicação grande.
* **Um microframework**: é um framework básico que fornece funcionalidades mínimas para criar aplicações Python simples.

Os mais populares são: Django, Flask, TurboGears e PyTorch.

**QUESTÕES**

### 1. Por que o Python é melhor que o Java?

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.

### 2. Quantos tipos de dados existem no Python?

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

### 3. Qual é a diferença entre uma 'tupla' e uma 'lista'?

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.

### 4. O que é 'decapagem' e 'desinteressante'?

**A decapagem** acontece quando um módulo dentro do Python é aceito e convertido em um módulo de string, e depois é despejado no arquivo. O **desengatamento** é quando você recupera o módulo de string do arquivo.

### 5. O que é 'lambda'?

**Lambda** é uma função anonimamente executada com apenas uma expressão única.

### 6. Como a memória é gerenciada no Python?

**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).

### 7. O que é 'passar'?

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

### 8. Você pode copiar um objeto em Python?

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

### 9. Como deletar um arquivo dentro do Python?

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

### 10. O que é um 'dicionário'?

Os dicionários são compostos de chaves e os valores correspondentes da chave. Aqui está um exemplo:

dict={'Car':'Ford','Type':'Mustang','Year':'1967'}  
print dict[Car]  
Ford  
print dict[Type]  
Mustang  
print dict[Year]  
1967

### 11. Python é uma linguagem interpretada?

Sim, o Python é uma linguagem de programação interpretada. 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.

### 12. Como o Python é orientado a objeto?

**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.

### 13. O que é 'fatiar'?

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