Linguagem de Programação

Python

RELATORIO

Python foi criado por Guido van Rossum e lançada pela primeira vez em 1991. A linguagem foi concebida como uma sucessora da linguagem ABC, com o objetivo de ser fácil de ler, escrever e manter. Os principais IDES são: Pycharm: que oferece recursos avançados, como ferramentas de controle de versão. Visual Studio Code: Vem oferecendo uma interface amigável e personalizável, além de uma comunidade ativa de desenvolvedores contribuindo para sua extensão. Eclipse: Tem recursos como autocompletamento, depuração, verificação de erros e suporte para bibliotecas, Jupyter Notebook: apliacação web que permite criar e compartilhar documentários que contém código Python, além de texto aplicativo e visualizações e Spyder, em diversas áreas sua aplicação é utilizada como desenvolvimento web, analise de dados e inteligência artificial, entre outras. As bibliotecas são conjuntos de código pré-compilado usadas que fornecem funcionalidades adicionais para facilitar o desenvolvimento de aplicação e para realizar tarefas específicas, como Numpy: biblioteca para manipulação de arrays e cálculos matemáticos, Pandas: biblioteca para manipulação e análise de dados, e Matplotlib: biblioteca para criação de gráficos e visualizacoes. Os principais Frameworks são: Django: um framework web completo e robusto, Flask: um framework web minimalista e flexível e Tensor Flow: um framework de machine learning e deep learning. Possuem ampla variedade de frameworks que facilitam o desenvolvimento de aplicações em diferentes áreas.

Questões

1.2: Porque o Python é melhor que o Java?

Python é mais fácil de abordar e brilha mais tarefas de ciência de dados e de aprendizagem automática.

1.3: Quantos tipos de dados existem no Python?

Em Python, há diversos tipos de dados que podem ser utilizados para representar diferentes tipos de valores e estruturas de dados. Os principais tipos de dados em Python incluem:

**->Números:** Inteiros (int) e números de ponto flutuante (float), usados para representar valores numéricos.

**->Booleanos:** Tipo de dado que pode ter apenas dois valores: True e False, usado para lógica condicional.

**->Listas (list)**: Coleção ordenada e mutável de itens.**Tuplas (tuple)**: Coleção ordenada e imutável de itens.

->**Strings (str)**: Sequência de caracteres

->**Conjuntos (set)**: Coleção não ordenada de elementos únicos.

**->Dicionários (dict)**: Coleção de pares chave-valor, onde cada chave é única e associada a um valor.

**->Tipos numéricos mais específicos:** Como números complexos (complex).

1.4: Qual é a diferença entre uma tupla e uma lista ?

A principal diferença é que uma lista é mutável (ou seja, pode ser modificada após a criação), enquanto uma tupla é imutável (não pode ser modificada após a criação). O que significa que seus elementos não podem ser alterados após a criação da tupla.

1.5: O que é 'decapagem' e 'desinteressante'?

**Decapagem:** Este termo poderia se referir à operação de remover caracteres especiais de uma string, como espaços em branco no início e no fim da string, utilizando métodos como .strip() em Python.

 **Desinteressante:** Em contexto de programação, isso poderia se referir a um código que não é relevante para a discussão atual, ou a uma operação que não causa impacto significativo ou interessante no código ou na execução do programa.

1.6: O que é lambada ?

Lambda em Python é frequentemente usadas em situações onde você precisa de uma função rápida para uma única tarefa e não quer definir uma função usando def. Elas são úteis principalmente quando você precisa passar uma função como argumento para outra função, como em operações de filtragem, mapeamento e ordenação de dados.

Por exemplo, ao usar funções como filter() e map()

1.7: Como a memória é gerenciada no Python?

Python utiliza um sistema de gerenciamento de memória automático, conhecido como coleta de lixo (garbage collection). Isso significa que o Python automaticamente gerencia o espaço de memória alocado para objetos que não são mais utilizados, liberando-os para serem reutilizados.

1.8: O que é 'passar'?

 Em Python, ‘pass’ é uma palavra-chave que serve como um espaço reservado para onde um comando é esperado, mas nenhuma ação é necessária. É frequentemente usado quando a sintaxe requer alguma instrução, mas o programador não quer executar nenhum código real.

1.9: Você pode copiar um objeto em Python?

Sim, é possível copiar objetos em Python. Isso pode ser feito utilizando a função copy() para objetos mutáveis como listas e dicionários, ou utilizando o módulo copy para copiar objetos mais complexos.

1.10: Como deletar um arquivo dentro do Python?

Para deletar um arquivo em Python, você pode usar a função os.remove() do módulo os.

import os os.remove('nome\_do\_arquivo.txt')

1.11: 0 que é um 'dicionário’?

Um dicionário em Python é uma estrutura de dados flexível e mutável que permite armazenar pares de chave-valor. Sao muito versáteis e são frequentemente utilizados para representar dados estruturados e fazer mapeamento entre chaves e valores.

1.12: Python é uma linguagem interpretada?

Sim, Python é uma linguagem de programação interpretada. Isso significa que o código fonte em Python não é compilado diretamente para código de máquina como em linguagens compiladas (como C++ ou Java), mas sim interpretado linha por linha por um interpretador Python. Isso torna Python mais flexível e facilita o desenvolvimento, pois os programas Python podem ser executados sem a necessidade de um passo explícito de compilação.

1.14: Como o Python é orientado a objeto?

Python facilita muito a compreensão de alguns conceitos, torna a programação , mais simples de ser entendida, facilita o trabalho em equipe, além de proporcionar um melhor controle e maior estabilidade na produção de grandes projetos.

1.15: 0 que é 'fatiar"?

'Fatiar' em Python refere-se à operação de acessar partes específicas de uma sequência (como uma lista, uma string, ou uma tupla) usando a sintaxe de fatiamento (slicing). Essa técnica permite que você extraia subconjuntos de elementos dessa sequência de maneira eficiente e concisa.

* sequencia: Pode ser uma lista, uma string, uma tupla, ou qualquer outra sequência indexada.
* início: O índice onde o fatiamento começa (incluído).
* fim: O índice onde o fatiamento termina (excluído).
* passo: O tamanho do passo entre os elementos (opcional).

ALUNO: Bianca Alves