**As 40 principais perguntas e respostas da entrevista em Python**

**1) O que é Python? Quais são os benefícios de usar o Python?**

*Python* é uma linguagem de programação com objetos, módulos, *threads*, exceções e gerenciamento de memória. Os benefícios de *pythons* são que é simples e fácil, portátil, estrutura de dados extensível e incorporada e é um código aberto.

**2) O que é PEP 8?**

PEP 8 é uma convenção de codificação, um conjunto de recomendações sobre como escrever seu código Python mais legível.

**3) O que é decapagem e decapagem?**

O módulo Pickle aceita qualquer objeto Python e o converte em uma representação de string e despeja em um arquivo usando a função dump, esse processo é chamado de decapagem. Enquanto o processo de recuperar objetos Python originais da representação de string armazenada é chamado de unpickling.

**4) Como o Python é interpretado?**

A linguagem Python é uma linguagem interpretada. O programa Python é executado diretamente da fonte código. Ele converte o código-fonte que é escrito pelo programador em um intermediário linguagem, que é novamente traduzida em linguagem de máquina que deve ser executada.

**5) Como a memória é gerenciada em Python?**

A memória do Python é gerenciada pelo espaço de heap privado do Python. Todos os objetos e dados do Python estruturas estão localizadas em um heap privado. O programador não tem acesso a esse heap e interpretador privado cuida desse heap privado do Python. A alocação de espaço de heap Python para objetos Python é feita pela memória Python Gerente. A API principal dá acesso a algumas ferramentas para o programador codificar. Python também tem um coletor de lixo embutido, que recicla toda a memória não utilizada e libera a memória e a disponibiliza para o espaço de heap.

**6) Quais são as ferramentas que ajudam a encontrar bugs ou realizar análises estáticas?**

PyChecker é uma ferramenta de análise estática que detecta os bugs no código-fonte Python e avisa

sobre o estilo e complexidade do bug. Pylint é outra ferramenta que verifica se o módulo atende ao padrão de codificação.

**7) O que são decoradores Python?**

Um decorador Python é uma mudança específica que fazemos na sintaxe Python para alterar funções facilmente.

**8) Qual é a diferença entre lista e tupla?**

A diferença entre lista e tupla é que a lista é mutável enquanto a tupla não é. Tupla pode ser hash para, por exemplo, como chave para dicionários.

**9) Como os argumentos são passados ​​por valor ou por referência?**

Tudo em Python é um objeto e todas as variáveis ​​contêm referências aos objetos. O os valores das referências estão de acordo com as funções; como resultado, você não pode alterar o valor do referências. No entanto, você pode alterar os objetos se for mutável.

**10) O que são as compreensões Dict e List?**

São construções de sintaxe para facilitar a criação de um Dicionário ou Lista com base em iterável.

**11) Quais são os tipos internos que o python fornece?**

Existem tipos internos mutáveis ​​e imutáveis ​​de Pythons Tipos internos mutáveis

Lista

Conjuntos

Dicionários

Tipos internos imutáveis

**12) O que é namespace em Python?**

Em Python, cada nome introduzido tem um lugar onde vive e pode ser fisgado. Isto é conhecido como namespace. É como uma caixa onde um nome de variável é mapeado para o objeto colocado.

Sempre que a variável for pesquisada, esta caixa será pesquisada, para obter o objeto correspondente.

**13) O que é lambda em Python?**

É uma função anônima de expressão única frequentemente usada como função inline.

**14) Por que as formas lambda em python não possuem declarações?**

Um formulário lambda em python não possui instruções, pois é usado para criar um novo objeto de função e, em seguida, devolvê-los em tempo de execução.

**15) O que é passar em Python?**

Pass significa, instrução Python sem operação, ou em outras palavras, é um espaço reservado em composto declaração, onde deve haver um espaço em branco e nada deve ser escrito lá.

**16) Em Python, o que são iteradores?**

Em Python, os iteradores são usados ​​para iterar um grupo de elementos, contêineres como list.

**17) O que é unittest em Python?**

Uma estrutura de teste de unidade em Python é conhecida como unittest. Ele suporta o compartilhamento de configurações, testes de automação, código de desligamento para testes, agregação de testes em coleções etc.

**18) Em Python o que é fatiar?**

Um mecanismo para selecionar um intervalo de itens de tipos de sequência como lista, tupla, strings etc. conhecido como fatiamento.

**19) O que são geradores em Python?**

A maneira de implementar iteradores é conhecida como geradores. É uma função normal, exceto que ele produz expressão na função.

**20) O que é docstring em Python?**

Uma string de documentação do Python é conhecida como docstring, é uma maneira de documentar o Python functions, modules and classes.

**21) Como você pode copiar um objeto em Python?**

Para copiar um objeto em Python, você pode tentar copy.copy() ou copy.deepcopy() para o caso geral. Você não pode copiar todos os objetos, mas a maioria deles.

**22) O que é índice negativo em Python?**

As sequências Python podem ser indexadas em números positivos e negativos. Para índice positivo, 0 é o primeiro índice, 1 é o segundo índice e assim por diante. Para índice negativo, (-1) é o último índice e (-2) é o penúltimo índice e assim por diante.

**23) Como você pode converter um número em uma string?**

Para converter um número em uma string, use a função embutida str(). Se você quer um octal ou

representação hexadecimal, use a função embutida oct() ou hex().

**24) Qual é a diferença entre Xrange e range?**

Xrange retorna o objeto xrange enquanto range retorna a lista e usa a mesma memória e

não importa o tamanho do intervalo.

**25) O que é módulo e pacote em Python?**

Em Python, módulo é a maneira de estruturar o programa. Cada arquivo de programa Python é um módulo, que importa outros módulos como objetos e atributos.

A pasta do programa Python é um pacote de módulos. Um pacote pode ter módulos ou subpastas.

**26) Mencione quais são as regras para variáveis ​​locais e globais em Python?**

Variáveis ​​locais: se uma variável recebe um novo valor em qualquer lugar dentro do corpo da função, é suposto ser local.

Variáveis ​​globais: aquelas variáveis ​​que são referenciadas apenas dentro de uma função são implicitamente global.

**27) Como você pode compartilhar variáveis ​​globais entre módulos?**

Para compartilhar variáveis ​​globais entre módulos em um único programa, crie um módulo especial.

Importe o módulo de configuração em todos os módulos de sua aplicação. O módulo estará disponível como variável global entre os módulos.

**28) Explique como você pode fazer um script Python executável no Unix?**

Para tornar um script Python executável no Unix, você precisa fazer duas coisas,

O modo do arquivo de script deve ser executável e a primeira linha deve começar com # ( #!/usr/local/bin/python)

**29) Explique como excluir um arquivo em Python?**

Usando um comando os.remove (nome do arquivo) ou os.unlink(nome do arquivo)

**30) Explique como você pode gerar números aleatórios em Python?**

Para gerar números aleatórios em Python, você precisa importar o comando como

importar aleatório random.random() Isso retorna um número de ponto flutuante aleatório no intervalo [0,1)

**31) Explique como você pode acessar um módulo escrito em Python a partir de C?**

Você pode acessar um módulo escrito em Python a partir de C seguindo o método,

Módulo = =PyImport\_ImportModule("");

**32) Mencione o uso do operador // em Python?**

É um operador Floor Division, que é usado para dividir dois operandos com o resultado como quociente mostrando apenas dígitos antes do ponto decimal. Por exemplo, 10//5 = 2 e 10,0//5,0 = 2,0.

**33) Mencione cinco benefícios de usar Python?**

Python é composto por uma enorme biblioteca padrão para a maioria das plataformas de Internet como Email, HTML, etc

Python não requer gerenciamento de memória explícito, pois o próprio interpretador aloca a memória para novas variáveis ​​e liberá-los automaticamente Fornece fácil legibilidade devido ao uso de colchetes Fácil de aprender para iniciantes Ter os tipos de dados integrados economiza tempo e esforço de programação ao declarar variáveis

**34) Mencione o uso da função split em Python?**

O uso da função split em Python é que ela quebra uma string em strings mais curtas usando o

separador definido. Ele fornece uma lista de todas as palavras presentes na string.

**35) Explique o que é Flask e seus benefícios?**

Flask é um micro framework web para Python baseado em “Werkzeug, Jinja 2 e boas intenções”

BSD licenciado. Werkzeug e jingja são duas de suas dependências.

Flask faz parte do micro-framework. O que significa que terá pouca ou nenhuma dependência de

bibliotecas externas. Isso torna a estrutura leve enquanto há pouca dependência para atualizar e

menos bugs de segurança.

**36) Mencione qual é a diferença entre Django, Pyramid e Flask?**

Flask é um “microframework” construído principalmente para um pequeno aplicativo com requisitos mais simples. Dentro frasco, você tem que usar bibliotecas externas. O frasco está pronto para uso.

Pyramid são construídos para aplicações maiores. Ele fornece flexibilidade e permite que o desenvolvedor use as ferramentas certas para o seu projeto. O desenvolvedor pode escolher o banco de dados, estrutura de URL, templates estilo e muito mais. A pirâmide é configurável pesada.

Assim como o Pyramid, o Django também pode ser usado para aplicações maiores. Inclui um ORM.

**37) Mencione o que é Flask-WTF e quais são suas características?**

Flask-WTF oferece integração simples com WTForms. Os recursos incluídos para o Flask WTF são

Integração com wtforms

Formulário seguro com token csrf

Proteção global de csrf

Integração de internacionalização

Recaptcha apoiando

Upload de arquivo que funciona com o Flask Uploads

**38) Explique qual é a maneira comum de o script Flask funcionar?**

A maneira comum para o script de frasco funcionar é

Ou deve ser o caminho de importação para seu aplicativo

Ou o caminho para um arquivo Python

**39) Explique como você pode acessar as sessões no Flask?**

Uma sessão basicamente permite que você se lembre de informações de uma solicitação para outra. Em um frasco, ele usa um cookie assinado para que o usuário possa ver o conteúdo da sessão e modificá-lo. O usuário pode modifique a sessão apenas se ela tiver a chave secreta Flask.secret\_key.

**40) Flask é um modelo MVC e se sim dê um exemplo mostrando o padrão MVC para o seu**

**aplicativo?**

Basicamente, o Flask é um framework minimalista que se comporta da mesma forma que o framework MVC. Então MVC é um ajuste perfeito para Flask, e o padrão para MVC que consideraremos para o exemplo a seguir