**Pythonインタビューの質問と回答**

**Pythonとは?**

Pythonは、最も広く使われている一般的なプログラミング言語の1つです。 1991年2月20日、グイド・ヴァン・ロッサムが開発した。 Pythonは無料です 非常にシンプルで清潔な構文でオープンソースの言語で、使いやすい 開発者が Python を学びます。 オブジェクト指向プログラミングをサポートし、最も 一般的に汎用プログラミングを実行するために使用される。 Python は複数 データサイエンス、機械学習、ディープラーニング、人工知能、 科学コンピューティングスクリプト, ネットワーキング, ゲーム開発 ウェブ開発、 ウェブスクレイピング、その他各種ドメイン

Pythonは、その単純性といくつかの実行能力のために指数関数的に成長します 数行のコードで機能します。 強力な計算をサポートする能力のため 強力なライブラリを使用して、さまざまなドメインで使用され、それが理由になります 世界中の Python 開発者にとって大きな需要。 企業は提供して喜んでいます これらの開発者に素晴らしいパッケージと利点。

Freshersの基本的なPythonインタビューの質問   
1。 Pythonとは? テクノロジーの世界における Python の人気のあるアプリケーションをいくつか紹介します。 Pythonは広く使われている汎用性の高い高レベルのプログラミング言語です。 作成されました 1991年、グイド・ファン・ロッサム氏より、Pythonソフトウェア財団が開発。 コードの読みやすさに重点を置いて設計され、その構文はプログラマーを可能にします コードの少ない行でコンセプトを表現する。

それはのために使用されます:   
システムスクリプト   
ウェブ開発   
ゲーム開発   
ソフトウェア開発   
複雑な数学   
2。 現状のシナリオのツールとして Python 言語を使用する利点は何ですか?

以下は、Python言語を使用する利点です。 オブジェクト指向言語   
レベルの高い言語

動的にタイプされた言語   
広範なサポート ライブラリー   
サードパーティモジュールの存在   
オープンソースとコミュニティ開発   
ポータブルとインタラクティブ   
ポータブル オペレーティングシステム   
3。 Pythonはコンパイルされた言語や通訳言語ですか?

実際には、Pythonは部分的にコンパイルされた言語であり、部分的に解釈された言語です。 ザ・オブ・ザ・ コードを実行するとコンパイル部分が最初に行われ、バイトコードが生成されます。 内部的にこのバイトコードは、Python仮想マシン(p.v.m)によって変換されます。 基盤となるプラットホーム(機械+操作システム)。

4。 Python で ‘#’ のシンボルは?

「#」は、行の後に来るすべてのコメントに使用されます。

5。 Mutableデータ型とImmutableデータ型の違いは何ですか? Mutable データ型は、実行時に変更できる i.e. を編集できます。 Eg – リスト、辞書、 など。

誤ったデータ型は、実行時に変更することはできません。 Eg – 文字列, ツプル、等。

6。 値で渡された引数やPythonの参照で渡された引数は?

Python のすべての変数はオブジェクトであり、全ての変数はオブジェクトへの参照を保持します。 ザ・オブ・ザ・ 参照値は関数に応じてあります。その結果、値を変更することはできません。 参照。 しかし、ミュータブルであればオブジェクトを変更することができます。

7。 セットと辞書の違いは何ですか?

set は、反復可能でミュータブルで重複しないデータ型の無秩序なコレクションです。 要素。

Python の辞書は、データ値の順序付けされたコレクションで、のようなデータ値を保存します。 サイトマップ

8。 リスト理解とは? 例を記述します。

リストの理解は、既存のリストに基づいてリストの作成を容易にする構文構造です 反復可能。

例:   
my\_list = [i は範囲(1, 10)]   
9月9日 ラムダ機能とは?

ラムダ関数は匿名関数です。 この機能は、任意の数を持つことができます。 パラメータは1つのステートメントだけを持つことができます。 例:   
a = ラムダ x, y : x\*y   
印刷(a(7、19))   
10月10日 Pythonのパスは何ですか?

パスとは、動作しない、つまり、化合物のプレースホルダです。 文、空白の左にあるべき場所と、そこに書かれているものは何もありません。

11月11日 Pythonの/と //の違いは何ですか?

/ は精密な分裂(result はフローティングポイント数)を表します。 分割(侮辱は整数)。 例:

5//2 = 2   
5月2日=2.5   
12月12日 Pythonで例外的な処理を行う方法は?

例外をキャッチするために使用される3つの主なキーワードがあります。 そしてそれに応じて回復のメカニズムを扱います。 試してみると、あるコードのブロック エラーを監視します。 エラーが発生した場合にブロックが実行されます。

最後のブロックの美しさは、エラーを試した後にコードを実行することです。 このブロックが取得 エラーが発生したか否かの誤りを実行します。 最後に、ブロックは オブジェクト/変数のクリーンアップ活動が必要です。

13 日 PythonでSwapcase関数とは何ですか?

すべての大文字を小文字やその逆文字に変換する文字列の関数です。 文字列の既存のケースを変更するために使用されます。 このメソッドは文字列のコピーを作成します。 スワップケースにすべての文字を含む。 例:

string = "GeeksforGeeks"

string.swapcase() ---> "gEEKSFORgEEKS"

14。 Pythonのループとループの違い

「for」ループは、一般的に様々なコレクションタイプの要素を介して反復するために使われます リスト、タプル、セット、辞書など。 開発者は、両方を持っている「for」ループを使用します 条件が始まり、終わります。 一方、ループは実際のループ機能です。 他のプログラミング言語で使われています。 プログラマはループ中に Python を使用します 終端条件を持つ場所。

15。 Pythonで関数を引数として渡すことはできますか?

はい、オブジェクト、変数(の)を含む関数に複数の引数を渡すことができます。 同じまたは異なるデータタイプ)、および機能。 関数は他の変数として渡すことができます オブジェクトだから関数。 より高い順序機能は、他の機能を取ることができる機能です 引数として関数。

もっと読むには、記事を参照してください。 Pythonで関数を引数に渡す

16。 \*argsと\*kwargsとは何ですか?

Python の関数に引数の変数数を渡すには、特別な構文を使う 機能仕様の\*argsおよび\*\*kwargs。 変数長さを渡るのに使用されています、 キーワードなしの引数リスト。 \* を使うと、\* と関連付ける変数が

反復可能で、それを反復し、より高い順序を使用することのような操作をすることができます 地図およびフィルターのような操作。

17。 Pythonでインデントが必要ですか?

Pythonではインデントが必要です。 Python通訳者は、グループに通知することができます ステートメントは Python のインデントを使用して、コードの特定のブロックに属します。 インデックス すべてのプログラミング言語で開発者が読みやすく、Pythonでは、 特定の順序でコードを刻むことは非常に重要です。

18.18.(日) Pythonのスコープとは?

変数を見つけることができる場所と、必要な場合は、その範囲がスコープと呼ばれます 変数の。

Python ローカル変数: ローカル変数は関数内で初期化されるものであり、 その機能にユニークです。 機能の外部からのアクセスはできません。

Python グローバル変数: グローバル変数は、定義され、外部に宣言されるもの 任意の関数は、任意の関数に指定されていません。

モジュールレベルの規模: 現在のモジュールのグローバルオブジェクトを参照します。 プログラム

最下の規模: プログラムが呼び出すことができる組み込みの名前を指します。 名前 参照されるのは、このスコープのオブジェクトの中で最後のものです。

19. 19.(日) Pythonのdocstringは何ですか?

Python のドキュメンテーション文字列(または docstrings)は、associating の便利な方法を提供します Python モジュール、関数、クラス、メソッドのドキュメント。

Docstringの宣言: docstrings は "'triple single quotes" または ""triple を使って宣言されます。 クラス、メソッド、関数宣言のすぐ下にある「二重引用」。 すべての関数は docstring を持っています。

Docstringsへのアクセス: docstring は \_\_doc\_\_ メソッドを使ってアクセスすることができます。 オブジェクトまたはヘルプ機能を使用する。

20 . . 動的にタイプされた言語は何ですか?

タイプされた言語は、データ型を定義する言語です。 コンパイル時に、またはランタイムで機械によって知られています。 タイプの言語は分類することができます 2つのカテゴリに:

静的タイプの言語: このタイプの言語では、変数のデータ型は、 プログラマが変数のデータ型を変数に指定しなければならないことを意味するコンパイル時間 宣言の時間。

動的にタイプされた言語: これらは、事前定義を必要としない言語です。 マシン自体でランタイムで解釈されるので、任意の変数のデータタイプ。 お問い合わせ 言語、通訳者は、その値に応じて実行時にデータ型を変数に割り当てます。 21. . Pythonで休憩、続行、パスは何ですか?

break ステートメントは、存在するループまたはステートメントを終了するために使用されます。 アフター つまり、コントロールは、ブレイクステートメントの後に存在するステートメントに渡します。 利用できる。

Continue は、ブレイクステートメントのようにループ制御ステートメントです。 継続声明 ブレークステートメントのその反対です, 代わりにループを語る, それは強制する ループの次の反復を実行します。

パスとは、動作しない、つまり、化合物のプレースホルダです。 文、空白の左にあるべき場所と、そこに書かれているものは何もありません。

22.(日) Pythonの組み込みデータ型は何ですか?

以下は、Python における標準または組み込みのデータタイプです。

数値: Python の数値型は数値値を持つデータを表します。 ツイート 数値は整数、浮動小数、ブール値、あるいは複雑な数値であってもよい。 シーケンスのタイプ: Python のシーケンスデータタイプは、類似または類似のオーダーコレクションです。 異なるデータタイプ。 Pythonにはいくつかのシーケンスタイプがあります。   
Python 文字列   
Pythonリスト

Python タプル   
Python の範囲   
マッピングの種類: Python では、ハッシュ可能なデータはランダムオブジェクトにマップできます。 マッピングオブジェクト。 現在、共通のマッピングタイプ、辞書のみ、 マッピングオブジェクトはミュータブルです。

Python辞書   
設定タイプ: Python では、Set は、反復可能でミュータブルなデータ型を並べ替えられたコレクションです。 重複要素がない。 セット内の要素の順序は未定義ですが、 様々な要素で構成されます。

23。 Pythonで数を埋める方法は?

Python math モジュールには、床を計算するために使用できるメソッドが含まれています。 番号。

Python の floor() メソッドは x よりも大きい整数ではなく x i.e のフロアを返します。 また、Python の ceil(x) メソッドは x i.e の天井値を返します。 x よりも大きいか等しい。

中間Pythonインタビュー質問   
24 . xrangeとレンジ関数の違いは何ですか?

range() と xrange() は、特定の数を反復するのに使用できる 2 つの関数です。 Python のループの時。 Python 3ではxrangeはありませんが、範囲関数は振る舞います Pythonのxrangeのような2.

範囲() – range() 関数を使って作成した数値のリストを返します。

xrange() – この関数は、数値を表示するために使用できるジェネレータオブジェクトを返します ループするだけで。 唯一の特定の範囲は、需要に表示され、したがって、レイジーと呼ばれる 評価。

25。 辞書の理解とは何ですか? 例を挙げる   
辞書の理解は辞書の作成を容易にする構文構造です 既存の反復可能に基づいて。

例:my\_dict = {i:1+7 範囲(1, 10)}

26. . Tupleの理解? もしそうなら、どうしても、どうしても、どうなのか?

(i 用 (1, 2, 3))   
ジェネレータで終わるため、PythonではTupleの理解は不可能です。 タプルの理解。

27.27.(日) リストとタプルの違いは?

リストとトゥプルの違いを分析してみましょう:

プロフィール

リストは Mutable データ型です。

リストはより多くのメモリを消費します   
リストは、インサートや削除などの操作を実行するのに適しています。

反復の影響は時間がかかります トゥプル

Tuples は Immutable データ型です。

Tuple はリストと比較して少ないメモリを消費します   
要素へのアクセスには、Tupleデータタイプが適しています。   
反復の影響は比較的高速です   
28.28.(日) 浅いコピーと深いコピーの違いは何ですか?

新規インスタンスタイプが作成されるとシャローコピーが使用されます。 すでにコピーされている値が深いコピーとしてコピーされます。

浅いコピーはプログラムの実行を高速化し、深いコイが遅くなります。

29. . sort() と sorted() python の関数によって、どのようなソート技術が使われていますか?

Python は Tim ソートアルゴリズムを使用してソートを行います。 最悪のケースが最も安定的なソートです。 O(NログN) 統合ソートとインサートソートから派生するハイブリッドソートアルゴリズムです。 実世界の多くのデータでよく実行するように設計されている。

30.30(日) デコレータとは?

デコレータは、特定の変更であるため、Pythonの非常に強力で便利なツールです。 Python の構文で機能を簡単に変更できます。

31.31. Pythonプログラムのデバッグ方法は?

このコマンドを使うと、Pythonプログラムをデバッグできます。 $ python -m pdb python スクリプト。 ログイン   
32。 Pythonのイテレータは何ですか?

Python では、イテレータを使用して、要素のグループ、リストのようなコンテナを反復します。 イテレータ アイテムのコレクションであり、リスト、タプル、または辞書にすることができます。 Pythonのイテレータ 保存された要素を反復するために \_\_itr\_\_ と next() メソッドを実行します。 一般的には、 Pythonでコレクション(リスト、タプル)を反復するループ。

33. . Pythonのジェネレータとは?

Pythonでは、ジェネレータは、イテレータの実装方法を指定する方法です。 それは普通です 関数以外の関数は関数内の式を yield します。 \_\_itr\_ を実行せず、 next() メソッドは他のオーバーヘッドも削減します。

関数が少なくとも yield ステートメントを含んでいると、ジェネレータになります。 収量キーワード 状態を保存して、現在の実行を一時停止し、同時に再開します。 お問い合わせ

34.34. Pythonは複数の継承をサポートしていますか?

Python は、Java とは異なり、複数の継承をサポートしています。 複数の継承とは、 複数の親クラスから派生できます。

35. 35. PythonのPolymorphismとは?

多形態主義とは、複数の形態を取る能力を意味します。 たとえば、親クラスの場合 ABCというメソッドを持ち、子クラスも同じ名前のメソッドを持つことができます。 独自のパラメータと変数を持つ ABC。 Pythonはポリモルフィズムを可能にします。

36 . Pythonでカプセル化を定義しますか?

カプセル化とは、コードとデータの結合を意味します。 Python クラスは例です。 カプセル化の。

37 . . Pythonでデータ抽象化を行うには?

Data Abstractionは、必要な詳細のみを提供し、実装を非表示にします。 ワールド。 インターフェイスと抽象的なクラスを使用して Python で実現できます。

38。 Pythonでメモリ管理を行う方法は?

Python は、メモリを管理するために、専用のヒープスペースを使用します。 基本的に、すべてのオブジェクトと 専用ヒープスペースにデータ構造を保存します。 プログラマがアクセスできない 通訳者がこの空間を大事にする空間です。 Python も組み込み すべての未使用のメモリをリサイクルし、記憶を解放する廃物のコレクター、 ヒープスペースにご利用いただけます。

39.39. Pythonを使用してファイルを削除する方法?

次のアプローチで Python を使用してファイルを削除できます。

os.remove()   
os.unlink()   
40 . Pythonでスライシングとは?

Python Slicing は、文字列の一部またはリストの一部を抽出するための文字列操作です。 この演算子では、スライシングを起動する場所、終了場所を指定して、 ステップ。 リストスライシングは、既存のリストから新しいリストを返します。

シンタックス: Lst[ 初期 : 終了 : インデックスジャンプ ]   
41. 41. Pythonの名前空間とは?

名前空間はネーミングを避けるために名前が一意であることを確認するために使用されるネーミングシステムです 紛争。

高度なPythonインタビューの質問と回答 42 . . PIPとは?

PIP は、Python インストーラ パッケージの略語で、シームレスなインターフェースをシームレスに提供します。 各種Pythonモジュールをインストールします。 パッケージを検索できるコマンドラインツールです。 インターネットはユーザのやり取りなしにインストールします。

43 . zip機能とは?

Python zip() 関数は zip オブジェクトを返します。これは複数のインデックスをマッピングします。 コンテナ。 それは反復可能をとり、それをイテレータに変換し、基づく要素を集約します 反復可能に渡される。 タプルのイテレータを返します。

44. 44. ピックリングとアンピクルスとは?

Pickle モジュールは任意の Python オブジェクトを受け入れ、それを文字列表現に変換します。 ダンプ関数を使用してファイルへダンプすると、このプロセスはピックリングと呼ばれます。 しばらくの間 保存された文字列表現から元の Python オブジェクトを取得するプロセスは、 unpickling と呼ばれる。

45. 45. Pythonでサルパッチングとは何ですか?

Pythonでは、サルパッチの用語は、クラスまたはモジュールの動的変更のみを参照します。 実行時に

# g.pyさん   
クラスの Geeks クラス:   
 def機能(自己):

印刷 "function()"

輸入m   
def サル\_function(自己):   
 "monkey\_function()" を印刷する

m.GeeksClass.function = サル\_function   
obj = m.Geeksクラス()   
obj.function()   
46。 Pythonの\_init\_\_()とは?

OOP 用語のコンストラクターと等価な \_\_init\_ は Python で予約されたメソッドです。 クラス。 新しいオブジェクトが開始されるたびに、\_init\_\_メソッドが自動的に呼び出されます。 このメソッドは、作成直後に新しいオブジェクトにメモリを割り当てます。 このメソッドは 変数を初期化するためにも使われます。

47. . 現在の時刻を表示するコードを書く?

currenttime= time.localtime(time.time())) の   
印刷(現時点では「現在の時間」)   
48。 Python の Access の指定子は何ですか?

Python は ‘\_’ のシンボルを使用して、特定のデータメンバーまたは A へのアクセス制御を決定します。 クラスのメンバー関数。 Python のクラスには 3 つの種類の Python アクセス修飾子があります。

公衆アクセスの修飾語: パブリック宣言されたクラスのメンバーは簡単に プログラムのどの部分からもアクセス可能です。 クラスのすべてのデータメンバーとメンバー関数 デフォルトではパブリックです。

保護されたアクセスの修飾語: 保護されるクラスのメンバーは、 由来するクラスへのアクセス クラスのすべてのデータメンバーは、 そのクラスのデータメンバーの前に、単一のアンダースコア ‘\_’ シンボルを追加します。

プライベートアクセス修飾子: プライベート宣言されたクラスのメンバーはアクセス可能 クラス内でのみ、プライベートアクセス修飾子は最も安全なアクセス修飾子です。 データデータ

前のダブルアンダースコア「\_\_」記号を追加することで、クラスのメンバーがプライベートで宣言されます。 そのクラスのデータメンバー。

49 . Pythonのユニットテストとは?

ユニットテストは、最も小さいテスト可能な部品が最も小さいソフトウェアテストの最初のレベルです ソフトウェアはテストされます。 これは、ソフトウェアの各ユニットが実行することを確認するために使用されます 設計しました。 ユニットテストフレームワークは、PythonのxUnitスタイルフレームワークです。 ホワイトボックス ユニットテストに用いられる試験方法

50. Pythonグローバル通訳ロック(GIL)?

Python Global Interpreter Lock(GIL)は、Pythonが使用するプロセスロックの一種です。 プロセスを扱うたびに。 一般的に、Pythonは1つのスレッドのみを使用して実行します。 記述文のセット。 単一スレッドプロセスとマルチのパフォーマンススレッドされたプロセスは Python で同じで、これは Python で GIL のからです。 お問い合わせ Pythonでマルチスレッドを実現しないため、グローバル通訳ロックが制限されています。 スレッドは単一スレッドとして機能します。

51。 Pythonで関数の注釈は何ですか?

関数のアノテーションは、関数パラメータにメタデータを追加できる機能です 値を返す。 これにより、関数パラメータの入力タイプを指定できます。 関数が返す値の戻りタイプ。

関数のアノテーションは、さまざまな部分に関連付けられている任意のPython式です 関数の。 これらの式はコンパイル時に評価され、Pythonの寿命がない場合 ランタイム環境。 Pythonは、これらの注釈に任意の意味を添付しません。 彼らは取る サードパーティのライブラリで解釈される場合、例えばmypy。

52 . Pythonで例外グループとは何ですか?

Python 3.11、Exceptionグループの最新機能。 ExceptionGroup は \* 以外の新しい構文を使ってください。 \* 記号は、複数の例外が使用できることを示します。 ※ 句を除き、それぞれに扱います。

ExceptionGroupは、例外の異なる種類のコレクション/グループです。 作成なし 複数の例外をグループ化することで、後で取得できるさまざまな例外をグループ化できます。 必要に応じて1つずつ、例外が保存される順序 例外グループでは、呼び出し時に問題ありません。

トライ:   
ExceptionGroup('Example ExceptionGroup') を上げる TypeError('Example TypeError'),   
ValueError('Example ValueError'),   
キーエラー('Example KeyError'),   
属性エラー('Example AttributeError')   
) )   
その他\* タイプ: エラー:   
・・・

その他\* バリュー eとしてエラー:   
・・・

e として例外\* (KeyError, AttributeError):   
・・・

53. . Pythonスイッチステートメントとは   
バージョン 3.10 以降では、Python は「structural」と呼ばれるスイッチケース機能を実装しました。 パターンマッチング。 マッチとケースのキーワードでこの機能を実装できます。 お問い合わせ アンダースコアシンボルは、スイッチステートメントのデフォルトケースを定義するために使用するものです。 Pythonで。

Note: Pythonの前に 3.10 Python はマッチステートメントをサポートしていません。

一致の言葉:   
 場合パターン-1:

アクション-1   
 ケースパターン2:   
 アクション-2   
 場合パターン-3:   
 アクション3   
 ケース \_:   
 アクション・デフォルト   
54。 ウォルラスオペレーターとは?

Walrus Operatorを使用すると、式内で値を変数に割り当てることができます。 お問い合わせ ループで複数回値を使用する必要があるときに便利ですが、 計算を繰り返します。

Walrus 演算子は `:=` の構文で表され、様々な で使用できる while ループと if ステートメントを含むコンテキスト。

Note: 3.8 以前の Python バージョンは、Walrus Operator をサポートしていません。

名前 = ["Jacob", "Joe", "Jim"]

if (name := input("name:") を名前に入力します: 印刷(f"Hello, {name}!")   
その他:   
 print("名前が見つかりません。")