

PYTHON INTERVIEW QUESTIONS

1) What is Python and what are the advantages of using Python?

-Python is a high-level, interpreted programming language known for its simplicity and readability. It supports a wide range of applications, from web development to data analysis and artificial intelligence.

Python offers advantages such as clear syntax, vast libraries, cross-platform compatibility, and a strong community, making it suitable for rapid development and diverse projects.

2) How do you comment in Python?

Use the "#" symbol for single-line comments and triple quotes (""" or """) for multi-line comments.

3) Explain the term "Pythonic" code.

Pythonic code follows the language's idioms and style conventions, emphasizing readability and simplicity.

4) What is a Python virtual environment?

A virtual environment isolates Python packages and dependencies, allowing developers to manage project-specific dependencies separately.

5) How do you create a function in Python?

Use the "def" keyword followed by the function name, parameters, and a colon. Define the function's body and optionally return a value.

6) Explain list comprehension in Python.

List comprehension is a concise way to create lists using a single line of code, iterating over an iterable and applying an expression.

7) What is PEP 8?

-PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable.

8) What is pickling and unpickling?

-Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

9) How Python is interpreted?

-Python language is an interpreted language. Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

10) How memory is managed in Python?

- Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
- The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
- Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

11) What are the tools that help to find bugs or perform static analysis?

-PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

12) What are Python decorators?

-A Python decorator is a specific change that we make in Python syntax to alter functions easily.

13) What is the difference between list and tuple?

The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries.

14) How are arguments passed by value or by reference?

-Everything in Python is an object and all variables hold references to the objects. The references values are according to the functions; as a result you cannot change the value of the references. However, you can change the objects if it is mutable.

15) What is Dict and List comprehensions are?

-They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable.

16) What are the built-in type does python provides?

-There are mutable and Immutable types of Python's built in types:

Mutable built-in types

- List
- Sets
- Dictionaries

Immutable built-in types

- Strings
- Tuples
- Numbers

17) What is namespace in Python?

-In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

18) What is lambda in Python?

-It is a single expression anonymous function often used as inline function.

19) Why lambda forms in python does not have statements?

-A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

20) What is pass in Python?

-Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

21) In Python what are iterators?

-In Python, iterators are used to iterate a group of elements, containers like list.

22) What is unit test in Python?

-A unit testing framework in Python is known as unit test. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

23) In Python what is slicing?

-A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

24) What are generators in Python?

-The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

25) What is docstring in Python?

-A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes.

26) What is the purpose of the "if name == 'main':" statement?

It ensures that the code within it runs only when the script is executed directly, not when imported as a module.

27) Differentiate between a tuple and a list in Python.

Tuples are immutable, ordered collections of elements, while lists are mutable and can be changed after creation.

28) How do you convert a string to an integer in Python?

Use the "int()" function to convert a string to an integer, for example, "int("123")".

29) What is the purpose of the "range()" function in Python?

The "range()" function generates a sequence of numbers within a given range, often used in loops.

30) How can you check if a given element is present in a list?

Use the "in" keyword to check if an element is present in a list, for example, "element in my_list".

31) Explain the difference between a shallow copy and a deep copy in Python.

A shallow copy creates a new object but references the same nested objects, while a deep copy creates entirely new objects for both the outer and nested objects.

32) How do you convert a string to lowercase or uppercase in Python?

Use the "lower()" method to convert a string to lowercase and the "upper()" method to convert it to uppercase.

33) What is the purpose of the "len()" function in Python?

The "len()" function returns the number of items in an iterable, such as a list or a string.

34) What is the purpose of the "if name == 'main':" statement?

It ensures that the code within it runs only when the script is executed directly, not when imported as a module.

35) Differentiate between a tuple and a list in Python.

Tuples are immutable, ordered collections of elements, while lists are mutable and can be changed after creation.

36) Explain the difference between deep copy and shallow copy in Python.

A shallow copy creates a new object but references the same nested objects. A deep copy creates entirely new objects for both the outer and nested objects.

37) What are decorators in Python?

Decorators are functions that modify the behavior of other functions. They are commonly used for code reuse and adding functionality to functions.

38) How can you handle exceptions in Python?

Use "try", "except", "else", and "finally" blocks to catch and handle exceptions, ensuring robust and error-tolerant code.

39) What is the Global Interpreter Lock (GIL) in Python?

The GIL is a mutex that allows only one thread to execute Python bytecode in a single process, limiting multi-core CPU utilization in certain scenarios.

40) Explain the usage of "with" statements in Python.

The "with" statement is used to simplify resource management, ensuring that resources like files or sockets are properly closed after usage.

41) How can you perform file I/O operations in Python?

Use the "open()" function with modes like "r" for reading, "w" for writing, and "a" for appending. Remember to close the file after usage.

42) What is the purpose of the "self" keyword in Python?

"self" is a reference to the instance of the class and is used to access class attributes and methods within the class.

43) How do you create a class in Python?

Use the "class" keyword followed by the class name and a colon. Define class methods and attributes within the class block.

44) Explain the concept of inheritance in Python.

Inheritance allows a class (subclass) to inherit properties and methods from another class (superclass), facilitating code reuse and hierarchy.

45) What is method overriding in Python?

Method overriding involves defining a method in a subclass with the same name and parameters as a method in the superclass, allowing the subclass to provide its implementation.

46) How do you implement multiple inheritance in Python?

Use a comma-separated list of base classes in the class definition to inherit attributes and methods from multiple parent classes.

47) What are modules and packages in Python?

Modules are single files containing Python code, while packages are directories containing modules and a special "init.py" file.

48) Explain the concept of a generator in Python.

A generator is a function that generates values on-the-fly using the "yield" keyword, saving memory by producing values one at a time.

49) How can you handle JSON data in Python?

Use the "json" module to serialize Python objects into JSON format and deserialize JSON data into Python objects.

50) What is the purpose of the "map()" function in Python?

The "map()" function applies a given function to each item in an iterable and returns an iterator containing the results.

51) Explain the concept of a set in Python.

A set is an unordered collection of unique elements. Sets are used for membership testing and eliminating duplicate values.

52) How can you reverse a string in Python?

Use string slicing with a step of -1 to reverse a string, for example, "string[::-1]".

53) What is the purpose of the "enumerate()" function in Python?

The "enumerate()" function adds a counter to an iterable, returning an iterator of tuples containing both the index and the value.

54) How do you remove duplicates from a list in Python?

Use the "set()" function to convert the list into a set and then back into a list to remove duplicates.

55) What is the purpose of the "zip()" function in Python?

The "zip()" function combines multiple iterables (lists, tuples, etc.) element-wise, creating an iterator of tuples.

56) Explain the concept of a dictionary in Python.

A dictionary is an unordered collection of key-value pairs, used for storing and retrieving data based on keys.

57) How can you iterate over a dictionary in Python?

Use "for key in dictionary" to iterate over keys or "for key, value in dictionary.items()" to iterate over key-value pairs.

58) What is the "pass" statement in Python used for?

The "pass" statement is a placeholder that does nothing, often used when a statement is syntactically required but no action is desired.

59) How do you find the length of a list in Python?

Use the "len()" function to find the number of elements in a list.

60) Explain the difference between "deep copy" and "shallow copy" of a list.

A shallow copy creates a new list but references the same objects, while a deep copy creates entirely new objects for both the list and its contents.

61) How can you reverse a list in Python?

Use list slicing with a step of -1, like "list[::-1]", to reverse the order of elements in a list.

62) What is the purpose of the "max()" and "min()" functions in Python?

The "max()" function returns the largest value in an iterable, while the "min()" function returns the smallest value.

63) How can you concatenate two lists in Python?

Use the "+" operator to concatenate two lists, for example, "list1 + list2".

64) Explain the "try" and "except" blocks in Python.

The "try" block contains code that might raise an exception, and the "except" block handles the exception by providing an alternative code path.

65) How do you convert a list to a tuple in Python?

Use the "tuple()" function to convert a list to a tuple, for example, "tuple(my_list)".

66) What is the purpose of the "join()" method in Python?

The "join()" method is used to concatenate elements of an iterable (e.g., a list) into a string using a specified separator.

67) How can you find the index of an element in a list in Python?

Use the "index()" method to find the index of a specified element in a list, for example, "my_list.index(element)".

68) Explain the concept of a generator in Python.

A generator is a special type of iterable that generates values one at a time using the "yield" keyword, saving memory and improving performance.

69) How do you create a dictionary in Python?

Use curly braces "{}" and key-value pairs separated by colons to create a dictionary, for example, "my_dict = {"key": "value"}".

70) What is the purpose of the "filter()" function in Python?

The "filter()" function filters elements from an iterable based on a provided function, returning an iterator containing the filtered items.

71) How can you remove an item from a list in Python?

Use methods like "remove()" to remove an element by value or "pop()" to remove an element by index from a list.

72) Explain the "with" statement in Python for file operations?

The "with" statement simplifies resource management by ensuring that resources like files are properly closed after usage.

73) How do you convert a number to a string in Python?

Use the "str()" function to convert a number to a string, for example, "str(123)".

74) What is the purpose of the "sorted()" function in Python?

The "sorted()" function returns a sorted list of elements from an iterable, leaving the original iterable unchanged.

75) How can you iterate over the keys and values of a dictionary in Python?

Use "for key in dictionary" to iterate over keys and "for key, value in dictionary.items()" to iterate over key-value pairs.

76) Explain the "continue" statement in Python.

The "continue" statement is used in loops to skip the current iteration and move to the next one.

77) How do you convert a tuple to a list in Python?

Use the "list()" function to convert a tuple to a list, for example, "list(my_tuple)".

78) What is the purpose of the "del" statement in Python?

The "del" statement is used to delete variables, objects, or elements from lists, dictionaries, and other data structures.

79) How can you convert a list of strings into a single string in Python?

Use the "join()" method with an empty string as the separator to concatenate the strings in the list into a single string.

80) Explain the "break" statement in Python.

The "break" statement is used in loops to immediately terminate the loop and exit its block of code.

81) How do you convert a string to a list of characters in Python?

Use list comprehension or the "list()" function to convert a string into a list of its individual characters.

82) What is the purpose of the "random" module in Python?

The "random" module provides functions for generating random numbers, sequences, and making random selections.

83) How can you find the highest occurrence of an element in a list in Python?

Use the "max()" function with the "key" argument set to "list.count" to find the element with the highest occurrence.

84) Explain the concept of a "generator expression" in Python.

A generator expression is a compact way to create generators using a similar syntax to list comprehensions.

85) How do you swap the values of two variables in Python?

Use tuple packing and unpacking or a temporary variable to swap the values of two variables.

86) What is the purpose of the "is" operator in Python?

The "is" operator checks whether two variables reference the same object in memory.

87) How can you convert a list of integers to a comma-separated string in Python?

Use the "join()" method to concatenate the integers as strings with commas in between.

88) Explain the concept of a "namespace" in Python.

A namespace is a container that holds a collection of identifiers (names) and their corresponding objects.

89) How do you copy a dictionary in Python?

Use the "copy()" method, the "dict()" constructor, or dictionary comprehension to create a copy of a dictionary.

90) What is the purpose of the "isinstance()" function in Python?

The "isinstance()" function is used to check if an object is an instance of a specific class or type.

91) How can you sort a list of dictionaries based on a specific key in Python?

Use the "sorted()" function with a custom key function or a lambda function to sort the list of dictionaries based on a specific key.

92) Explain the "pass" statement in Python.

The "pass" statement is a placeholder that does nothing. It is often used to create minimal implementations or for syntactic purposes.

93) How do you concatenate two strings in Python?

Use the "+" operator to concatenate two strings, for example, "string1 + string2".

94) What is the purpose of the "split()" method in Python?

The "split()" method splits a string into a list of substrings based on a specified delimiter.

95) How can you find the smallest element in a list in Python?

Use the "min()" function to find the smallest element in a list.

96) Explain the concept of a "closure" in Python.

A closure is a function object that remembers values in the enclosing scope even if they are not present in memory.

97) How do you create a tuple in Python with a single element?

Add a comma after the element, like "(element,)" to create a tuple with a single element.

98) What is the purpose of the "reduce()" function in Python?

The "reduce()" function applies a given function to the elements of an iterable in a cumulative way, reducing the iterable to a single value.

99) How can you convert a dictionary to a list of tuples in Python?

Use the "items()" method to convert the key-value pairs of a dictionary into a list of tuples.

100) Explain the concept of a "default dictionary" in Python.

A default dictionary is a subclass of the "dict" class that automatically creates default values for missing keys.

101) How do you find the highest value of a dictionary in Python?

Use the "max()" function with the "key" argument set to "dict.get" to find the key with the highest value in a dictionary.

102) What is the purpose of the "chr()" function in Python?

The "chr()" function returns a string representing a character whose Unicode code point matches the given integer.

103) How can you remove all occurrences of a value from a list in Python?

Use a list comprehension to create a new list without the specified value.

104) Explain the concept of a "private variable" in Python.

Python does not have true private variables, but variables with a name prefixed by an underscore (e.g., "_variable") are considered private by convention.

105) How do you create a set in Python?

Use curly braces "{}" or the "set()" constructor to create a set, for example, "my_set = {1, 2, 3}".

106) What is the purpose of the "len()" function in Python?

The "len()" function returns the number of items in an iterable, such as a list or a string.

107) How can you find the index of the first occurrence of a substring in a string in Python?

Use the "index()" method to find the index of the first occurrence of a substring.

108) Explain the concept of a "docstring" in Python.

A docstring is a string that provides documentation and describes the purpose and usage of a function, module, class, or method.

109) How can you remove white spaces from both ends of a string in Python?

Use the "strip()" method to remove white spaces from both the beginning and end of a string.

110) What is the purpose of the "re" module in Python?

The "re" module provides functions for working with regular expressions, allowing you to perform advanced string manipulation.

111) How can you reverse the order of words in a string in Python?

Use string splitting and joining to reverse the order of words in a string, like "reversed_words = ' '.join(original_string.split()[::-1])".

112) Explain the concept of a "list comprehension" in Python.

A list comprehension is a concise way to create lists using a single line of code, applying an expression to each item in an iterable.

113) What is the purpose of the "sum()" function in Python?

The "sum()" function returns the sum of all items in an iterable, such as a list or tuple.

114) How can you remove a specific character from a string in Python?

Use the "replace()" method to replace occurrences of a specific character with an empty string.

115) Explain the concept of "polymorphism" in Python.

Polymorphism allows objects of different classes to be treated as objects of a common superclass, facilitating code reusability and flexibility.

116) What is the purpose of the "all()" and "any()" functions in Python?

The "all()" function returns True if all elements in an iterable are True, while the "any()" function returns True if any element is True.

117) How can you format a string in Python?

Use the "format()" method or f-strings (formatted string literals) to insert values into placeholders within a string.

118) Explain the concept of "deep copy" and "shallow copy" in Python.

A shallow copy creates a new object but references the same nested objects, while a deep copy creates entirely new objects for both the outer and nested objects.

119) How can you capitalize the first letter of a string in Python?

Use the "capitalize()" method to capitalize the first letter of a string.

120) What is the purpose of the "enumerate()" function in Python?

The "enumerate()" function adds a counter to an iterable, returning an iterator of tuples containing both the index and the value.

121) Explain the concept of a "set comprehension" in Python.

A set comprehension is a concise way to create sets using a single line of code, applying an expression to each item in an iterable.

122) How can you check if a string starts with a specific substring in Python?

Use the "startswith()" method to check if a string starts with a specific substring.

123) What is the purpose of the "round()" function in Python?

The "round()" function rounds a floating-point number to the specified number of decimal places.

124) Explain the concept of "name mangling" in Python.

Name mangling is a mechanism to make class variables less prone to name conflicts by adding a prefix to their names.

125) How can you find the lowest common multiple (LCM) of two numbers in Python?

Use the "math.gcd()" function to find the greatest common divisor (GCD) and calculate the LCM using the formula: $LCM(a, b) = (a * b) / GCD(a, b)$.

126) What is the purpose of the "ord()" function in Python?

The "ord()" function returns the Unicode code point of a specified character.

127) How can you remove leading white spaces from a string in Python?

Use the "lstrip()" method to remove leading white spaces from a string.

128) Explain the concept of "method resolution order" (MRO) in Python.

The method resolution order is the sequence in which Python searches for methods and attributes in classes, following the C3 linearization algorithm.

129) What is the purpose of the "filter()" function in Python?

The "filter()" function filters elements from an iterable based on a provided function, returning an iterator containing the filtered items.

130) How can you check if a string ends with a specific substring in Python?

Use the "endswith()" method to check if a string ends with a specific substring.

131) Explain the concept of a "dictionary comprehension" in Python.

A dictionary comprehension is a concise way to create dictionaries using a single line of code, applying an expression to each item in an iterable.

132) What is the purpose of the "divmod()" function in Python?

The "divmod()" function returns the quotient and remainder of the division of two numbers.

133) How can you convert a string to a floating-point number in Python?

Use the "float()" function to convert a string to a floating-point number, for example, "float("3.14")".

134) Explain the concept of "multiple inheritance" in Python.

Multiple inheritance allows a class to inherit attributes and methods from multiple parent classes, creating complex class hierarchies.

135) What is the purpose of the "callable()" function in Python?

The "callable()" function checks if an object is callable (i.e., if it can be invoked as a function).

136) How can you check if all characters in a string are alphabetic in Python?

Use the "isalpha()" method to check if all characters in a string are alphabetic.

137) Explain the concept of "method chaining" in Python.

Method chaining involves calling multiple methods on an object in a single line of code, allowing for concise and sequential operations.

138) What is the purpose of the "zip()" function in Python?

The "zip()" function combines multiple iterables (lists, tuples, etc.) element-wise, creating an iterator of tuples.

139) How can you count the occurrences of a specific element in a list in Python?

Use the "count()" method to count the occurrences of a specific element in a list.

140) Explain the concept of "list slicing" in Python.

List slicing allows you to create a new list by extracting a subset of elements from an existing list using specified start, stop, and step indices.

141) What is the purpose of the "map()" function in Python?

The "map()" function applies a given function to each item in an iterable and returns an iterator containing the results.

142) How can you convert a string to an integer in Python?

Use the "int()" function to convert a string to an integer, for example, "int("123")".

143) Explain the concept of "tuple packing" and "tuple unpacking" in Python.

Tuple packing involves creating a tuple by grouping values together, while tuple unpacking involves assigning individual values to variables from a tuple.

144) What is the purpose of the "next()" function in Python?

The "next()" function retrieves the next item from an iterator. It can also take a second argument to provide a default value if the iterator is exhausted.

145) How can you capitalize all words in a string in Python?

Use the "title()" method to capitalize the first letter of each word in a string.

146) Explain the concept of a "generator expression" in Python.

A generator expression is a concise way to create generators using a similar syntax to list comprehensions.

147) What is the purpose of the "pow()" function in Python?

The "pow()" function calculates the power of a number with a specified exponent, optionally using a third argument as a modulo.

148) How can you convert a floating-point number to an integer in Python?

Use the "int()" function to convert a floating-point number to an integer, truncating the decimal part.

149) Explain the concept of "ternary operator" in Python.

The ternary operator is a shorthand way to write conditional expressions, with the syntax: "value_if_true if condition else value_if_false".

150) What is the purpose of the "sum()" function in Python?

The "sum()" function returns the sum of all items in an iterable, such as a list or tuple.

151) How can you count the occurrences of a substring in a string in Python?

Use the "count()" method to count the occurrences of a substring in a string.

152) Explain the concept of "list comprehension" in Python.

A list comprehension is a concise way to create lists using a single line of code, applying an expression to each item in an iterable.

153) What is the purpose of the "enumerate()" function in Python?

The "enumerate()" function adds a counter to an iterable, returning an iterator of tuples containing both the index and the value.

154) How can you reverse the order of elements in a list in Python?

Use list slicing with a step of -1, like "reversed_list = my_list[::-1]", to reverse the order of elements.

155) Explain the concept of "deep copy" and "shallow copy" in Python.

A shallow copy creates a new object but references the same nested objects, while a deep copy creates entirely new objects for both the outer and nested objects.

156) What is the purpose of the "join()" method in Python?

The "join()" method concatenates elements of an iterable (e.g., a list) into a single string using a specified separator.

157) How can you remove leading and trailing white spaces from a string in Python?

Use the "strip()" method to remove leading and trailing white spaces from a string.

158) Explain the concept of "method chaining" in Python.

Method chaining involves calling multiple methods on an object in a single line of code, allowing for concise and sequential operations.

159) What is the purpose of the "sorted()" function in Python?

The "sorted()" function returns a sorted list of elements from an iterable, leaving the original iterable unchanged.

160) How can you reverse a string in Python?

Use string slicing with a step of -1 to reverse a string, like "reversed_string = string[::-1]".

161) Explain the concept of a "dictionary comprehension" in Python.

A dictionary comprehension is a concise way to create dictionaries using a single line of code, applying an expression to each item in an iterable.

162) What is the purpose of the "eval()" function in Python?

The "eval()" function evaluates a Python expression from a string and returns the result.

163) How can you check if a string is numeric in Python?

Use the "isnumeric()" method to check if a string consists of only numeric characters.

164) Explain the concept of "namespace" in Python.

A namespace is a container that holds a collection of identifiers (names) and their corresponding objects.

165) What is the purpose of the "len()" function in Python?

The "len()" function returns the number of items in an iterable, such as a list or a string.

166) How can you convert a list of strings into a single string in Python?

Use the "join()" method with an empty string as the separator to concatenate the strings in the list into a single string.

167) Explain the concept of "set comprehension" in Python.

A set comprehension is a concise way to create sets using a single line of code, applying an expression to each item in an iterable.

168) How can you copy an object in Python?

-To copy an object in Python, you can try `copy.copy()` or `copy.deepcopy()` for the general case. You cannot copy all objects but most of them.

169) What is negative index in Python?

-Python sequences can be indexed with positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

170) How you can convert a number to a string?

-In order to convert a number into a string, use the inbuilt function `str()`. If you want an octal or hexadecimal representation, use the inbuilt function `oct()` or `hex()`.

171) What is the difference between Xrange and range?

-Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

172) What is module and package in Python?

-In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes. The folder of Python program is a package of modules. A package can have modules or subfolders.

173) Mention what are the rules for local and global variables in Python?

-Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.

Global variables: Those variables that are only referenced inside a function are implicitly global.

174) How can you share global variables across modules?

-To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

175) Explain how can you make a Python Script executable on Unix?

-To make a Python Script executable on Unix, you need to do two things,

- Script file's mode must be executable and
- the first line must begin with

176) Explain how to delete a file in Python?

-By using a command `os.remove(filename)` or `os.unlink(filename)`

177) Explain how can you generate random numbers in Python?

-To generate random numbers in Python, you need to import command as `import random` `random.random()` This returns a random floating-point number in the range `[0,1)`

178) Explain how can you access a module written in Python from C?

You can access a module written in Python from C by following method, `Module = PyImport_ImportModule("");`

179) Mention what is the difference between Django, Pyramid, and Flask?

-Flask is a “microframework” primarily build for a small application with simpler requirements. In flask, you have to use external libraries. Flask is ready to use.

Pyramid are built for larger applications. It provides flexibility and lets the developer use the right tools for their project. The developer can choose the database, URL structure, templating style and more. Pyramid is heavy configurable.

Like Pyramid, Django can also use for larger applications. It includes an ORM.

180) Mention the use of the split function in Python?

-The use of the split function in Python is that it breaks a string into shorter strings using the defined separator. It gives a list of all words present in the string.

181) Mention five benefits of using Python?

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.
- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically
- Provide easy readability due to use of square brackets
- Easy-to-learn for beginners
- Having the built-in data types saves programming time and effort from declaring variables.

182) Mention the use of // operator in Python?

-It is a Floor Division operator, which is used for dividing two operands with the result as quotient showing only digits before the decimal point. For instance, $10//5 = 2$ and $10.0//5.0 = 2.0$

183) What is the difference between Python 2.x and Python 3.x?

Python 3.x is the latest version with improved features and syntax.

184) What is indentation in Python?

Indentation is used to define block-level structure.

185) What are variables in Python?

Variables are names given to memory locations.

186) What are data types in Python?

Int, float, string, list, tuple, dictionary, set.

187) What are operators in Python?

Arithmetic, comparison, logical, assignment, bitwise.

188) What are control structures in Python?

If-else, for, while.

189) What are functions in Python?

Functions are reusable blocks of code.

190) What are modules in Python?

Modules are pre-written code libraries.

191) What is object-oriented programming (OOP) in Python?

OOP is a programming paradigm based on objects and classes.

192) What is the difference between static and dynamic typing in Python?

Python is dynamically typed.

193) What is the Global Interpreter Lock (GIL) in Python?

GIL is a mechanism to prevent multiple threads from executing Python bytecodes at once.

194) How does memory management work in Python?

Python uses automatic memory management through a garbage collector.

195) How do you handle file input/output in Python?

Using built-in functions like open(), read(), write(), close().

196) What is the purpose of the with statement in Python?

with statement is used for exception handling and resource management.

197) How do you create a thread in Python?

Using threading module.

198) What is the difference between join() and detach() methods in Python?

join() waits for the thread to finish, detach() doesn't wait.

199) How do you create a process in Python?

Using multiprocessing module.

200) What is the purpose of the if __name__ == "__main__" block in Python?

It ensures that the code under it runs only when the script is run directly.

201) How do you create a socket in Python?

Using socket module.

202) What is the difference between TCP and UDP sockets in Python?

TCP is connection-oriented, UDP is connectionless.

203) How do you create a web server in Python?

Using http.server module.

204) What is the purpose of the requests library in Python?

requests library is used for making HTTP requests.

205) How do you create a GUI application in Python?

Using tkinter or PyQt libraries.

206) What is the purpose of the unittest module in Python?

unittest module is used for unit testing.

207) What is the purpose of the pip package manager in Python?

pip is used for installing and managing packages.

208) How do you create a virtual environment in Python?

Using venv module.

209) How do you create a class with abstract methods in Python?

Using abc module.

210) What is the purpose of the property decorator in Python?

property decorator is used for creating getters and setters.

211) How do you create a context manager in Python?

Using contextlib module.

212) What is the purpose of the try-except-finally block in Python?

It is used for exception handling.

213) How do you create a generator expression in Python?

Using () instead of [].

214) What is the purpose of the next() function in Python?

next() function is used for retrieving the next item from an iterator.

215) How do you create a dictionary comprehension in Python?

Using { } with a for loop.

216) What is the purpose of the zip() function in Python?

zip() function is used for combining multiple iterables.

217) How do you create a set comprehension in Python?

Using { } with a for loop.

218) What is the purpose of the enumerate() function in Python?

enumerate() function is used for looping over iterables with indices.

219) How do you create a list comprehension in Python?

Using [] with a for loop.

220) What is the purpose of the sorted() function in Python?

sorted() function is used for sorting iterables.

221) How do you create a tuple comprehension in Python?

Using () with a for loop.

222) What is the purpose of the re module in Python?

re module is used for regular expressions.

223) How do you create a regular expression pattern in Python?

Using re module.

224) What is the purpose of the match() function in Python?

match() function is used for matching regular expressions.

225) How do you create a group in regular expressions in Python?

Using ().

226) What is the purpose of the findall() function in Python?

findall() function is used for finding all occurrences of a pattern.

227) How do you create a named group in regular expressions in Python?

Using (?P<name>pattern).

228) What is the purpose of the sub() function in Python?

sub() function is used for replacing occurrences of a pattern.

229) How do you create a compiled regular expression pattern in Python?

Using re.compile().

230) What is the purpose of the split() function in Python?

split() function is used for splitting strings.

231) What is the purpose of the hash() function in Python?

hash() function is used for getting the hash value of an object.

232) How do you create a string with a specific encoding in Python?

Using encode() function.

233) What is the purpose of the bytes() function in Python?

bytes() function is used for creating a bytes object.

234) How do you create a string with a specific decoding in Python?

Using decode() function.

235) What is the purpose of the ord() function in Python?

ord() function is used for getting the Unicode code point of a character.

236) How do you create a string with a specific Unicode code point in Python?

Using chr() function.

237) What is the purpose of the ascii() function in Python?

ascii() function is used for getting the ASCII representation of an object.

238) How do you create a string with a specific ASCII value in Python?

Using chr() function.

239) What is the purpose of the repr() function in Python?

repr() function is used for getting the representation of an object.

240) How do you create a string with a specific representation in Python?

Using repr() function.

241) How do you create a string in Python?

Using quotes or triple quotes.

242) What is the purpose of the format() function in Python?

format() function is used for formatting strings.

243) How do you create a byte string in Python?

Using b prefix.

244) What is the purpose of the encode() function in Python?

encode() function is used for encoding strings.

245) How do you create a Unicode string in Python?

Using u prefix.

246) What is the purpose of the decode() function in Python?

decode() function is used for decoding strings.

247) How do you create a raw string in Python?

Using r prefix.

248) What is the purpose of the repr() function in Python?

repr() function is used for getting the representation of an object.

249) How do you create a multiline string in Python?

Using triple quotes.

250) What is the purpose of the str() function in Python?

str() function is used for converting an object to a string.

251) How do you create a string with escape sequences in Python?

Using backslash.

252) What is the purpose of the len() function in Python?

len() function is used for getting the length of a string.

253) How do you create a substring in Python?

Using slicing.

254) What is the purpose of the lower() function in Python?

lower() function is used for converting a string to lowercase.

255) How do you create a string with uppercase letters in Python?

Using upper() function.

256) What is the purpose of the title() function in Python?

title() function is used for converting a string to title case.

257) How do you create a string with swapped cases in Python?

Using swapcase() function.

256) What is the purpose of the strip() function in Python?

strip() function is used for removing leading and trailing whitespaces.

257) How do you create a string with removed whitespaces in Python?

Using replace() function.

258) What is the purpose of the split() function in Python?

split() function is used for splitting a string.

259) How do you create a string with joined substrings in Python?

Using join() function.

260) What is the purpose of the find() function in Python?

find() function is used for finding a substring.

261) How do you create a string with replaced substrings in Python?

Using replace() function.

262) What is the purpose of the count() function in Python?

count() function is used for counting occurrences of a substring.

263) How do you create a string with concatenated strings in Python?

Using + operator.

264) What is the purpose of the in operator in Python?

in operator is used for checking if a substring is present.

265) How do you create a string with repeated strings in Python?

Using * operator.

266) What is the purpose of the is operator in Python?

is operator is used for checking if two strings are the same object.

267) How do you create a string with formatted values in Python?

Using format() function.

268) What is the purpose of the f string notation in Python?

f string notation is used for formatting strings.

269) How do you create a string with escaped characters in Python?

Using backslash.

270) What is the purpose of the chr() function in Python?

chr() function is used for getting the character represented by a Unicode code point.

271) How do you create a string with Unicode characters in Python?

Using Unicode code points.

272) What is the purpose of the ord() function in Python?

ord() function is used for getting the Unicode code point of a character.

273) How do you create a string with ASCII characters in Python?

Using ASCII values.

274) What is the purpose of the ascii() function in Python?

ascii() function is used for getting the ASCII representation of an object.

275) How do you create a string with hexadecimal characters in Python?

Using hexadecimal values.