**1. What are different data**[**types in Python**](https://www.cyberithub.com/inheritance-concepts-and-its-5-different-types-in-python-with-examples/)**Collection ?**

**Ans. There are four different types:-**

* **List**
* **Tuple**
* **Set**
* **Dictionary**

**2. Does tuple collection type allows duplicate values ?**

**Ans. Yes**

**3. Which is an unordered and unindexed collection**[**type in Python**](https://www.cyberithub.com/shallow-and-deep-types-of-copy-in-python-with-best-examples/)**?**

**Ans. Set**

**4. Does Set allows duplicate values ?**

**Ans. No**

**5. Which is an ordered and changeable collection type in Python ?**

**Ans. List**

### ****6. Does Dictionary Collection Types allows duplicate value ?****

**Ans. No**

### ****7. Can we****[**change values**](https://www.cyberithub.com/change-default-umask-values-permanently/)****in Tuple Collection Type ?****

**Ans. No**

### ****8. Which function can be used to determine the number of members in tuple collection type ?****

**Ans. len() function**

### ****9. Which method is used to update the items in Python Dictionary ?****

**Ans. update() method**

### ****10. Which method can be used to add items from another set to current set ?****

**Ans. update() method**

**11. What are Arithmetic Operators? What are various types of arithmetic operators that we can use in Python?  
Answer:  
Arithmetic operators are used to performing mathematical functions such as addition, subtraction, division, and multiplication.**

**12. What is the Arithmetic operator’s precedence in Python?  
Answer:  
When more than one arithmetic operator appears in an expression the operations will execute in a specific order. In Python, the operation precedence follows as per the acronym PEMDAS.**  
**Parenthesis  
Exponent**  
**Multiplication  
Division  
Addition  
Subtraction**

**13. What are relational operators?  
Answer:  
Relational operators are also known as conditional or comparison operators. Relational operators in Python are defined as follows:**

1. **==: returns true if two operands are equal**
2. **!=: returns true if two operands are not equal**
3. **>: returns true if the left operand is greater than the right operand**
4. **<: returns true if the left operand is smaller than the right operand**
5. **>=: returns true if the left operand is greater than or equal to the right operand**
6. **<=: returns true if the left operand is smaller or equal to the right operand**

**14. What are logical operators?**  
**Answer:  
Logical operators are generally used in control statements like if and while. They are used to control program flow. The logical operator evaluates a condition and returns “True” or “False” depending on whether the condition evaluates to True or False. Three logical operators in Python are as follows:**

* **‘and’**
* **‘or’ and**
* **‘not’**

**15. What are bitwise operators?**  
**Answer:  
Bitwise operators work on bits and perform bit-by-bit operations. In Python, the following bit-wise operations are defined:  
AND – &**  
2 & 3  
  
**OR-|**  
2|3  
 **One’s complement – ~**  
>>> ~2  
-3  
**XOR -∧**  
2∧3

**Right shift ->>**  
2>>2  
**Left shift -<<**  
2<<2

**16.** [**What is pass in Python?**](https://www.360digitalgyan.com/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.**

**17.** [**What Does The Continue Do In Python?**](https://www.360digitalgyan.com/what-does-the-continue-do-in-pythonnbsp)

**The continue is a jump statement in Python which moves the control to execute the next iteration in a loop leaving all the remaining instructions in the block unexecuted.**

**The continue statement is applicable for both the “while” and “for” loops.**

**18.**[**When Should You Use The “Break” In Python?**](https://www.360digitalgyan.com/when-should-you-use-the-ldquobreakrdquo-in-pythonnbsp)

**Python provides a break statement to exit from a loop. Whenever the break hits in the code, the control of the program immediately exits from the body of the loop.**

**The break statement in a nested loop causes the control to exit from the inner iterative block.**

[**19. What Is The Difference Between Pass And Continue In Python?**](https://www.360digitalgyan.com/what-is-the-difference-between-pass-and-continue-in-pythonnbsp)

**The continue statement makes the loop to resume from the next iteration.**

**On the contrary, the pass statement instructs to do nothing, and the remainder of the code executes as usual.**

**20.**[**What Is A Built-In Function That Python Uses To Iterate Over A Number Sequence?**](https://www.360digitalgyan.com/what-is-a-built-in-function-that-python-uses-to-iterate-over-a-number-sequence)

**Range() generates a list of numbers, which is used to iterate over for loops.**

**The range() function accompanies two sets of parameters.**

**range(stop)**

**stop: It is the no. of integers to generate and starts from zero. eg. range(3) == [0, 1, 2].**

**range([start], stop[, step])**

**Start: It is the starting no. of the sequence.**

**Stop: It specifies the upper limit of the sequence.**

**Step: It is the incrementing factor for generating the sequence.**

**21. Can you explain what a for loop is?**

**A for loop is a type of loop that helps you run a certain set of commands over and over again. The for loop will keep track of how many times it has run the commands, and it will stop running the commands once it reaches a certain number.**

**22. What are the different ways of iterating over a list using a for loop?**

**The most common way to iterate over a list is using a for loop. However, there are a few different ways to do this. One way is to use a for loop with a range. This will allow you to specify the start and end points of the list. Another way is to use a for loop with an enumerate. This will give you the index of each element in the list as you iterate over it. Finally, you can also use a for loop with a zip. This will allow you to iterate over multiple lists at the same time.**

**23. When would you use a break statement inside a for loop?**

**A break statement is used to immediately exit out of a loop. This can be useful if you have a loop that is running through a large number of items and you want to exit out of the loop as soon as a certain condition is met.**

**24. How many types of loops are there in Python? Can you give me some examples of each type of loop?**

**There are three types of loops in Python: the for loop, the while loop, and the nested loop. The for loop is used to iterate through a sequence, such as a list or a string. The while loop is used to execute a block of code as long as a condition is true. The nested loop is used to execute a block of code multiple times.**

**25. What are the two types of functions in Python?**

**There are two types of functions in Python: built-in functions and user-defined functions. Built-in functions are functions that are already defined in the Python language, such as the print() function. User-defined functions are functions that are created by the user, and they can be created to do anything that the user wants them to do.**

**26.Do Python functions have return values? If yes, then how many can they have?**

**Yes, Python functions can have return values. They can have a single return value, or they can have multiple return values.**

**27. Why does Python support both positional and keyword arguments to its functions?**

**Python supports both positional and keyword arguments in order to give developers more flexibility when designing their functions. Positional arguments are those that are passed in by position, without explicitly specifying the parameter name. Keyword arguments are those that are passed in by explicitly specifying the parameter name. Python allows for both types of arguments so that developers can choose the approach that makes the most sense for their particular function.**

**28. Is there any way to define static methods in Python? If yes, then how?**

**Static methods are defined in Python by using the @staticmethod decorator. This decorator can be applied to any method, and will cause the method to be treated as a static method, even if it is not defined as such.**

**29. Is it possible to pass a variable number of arguments to a function in Python? If yes, then how?**

**Yes, it is possible to pass a variable number of arguments to a function in Python. This can be done using the \*args and \*\*kwargs parameters. \*args allows for a variable number of non-keyworded arguments to be passed to a function, while \*\*kwargs allows for a variable number of keyworded arguments to be passed.**

**30. Can you explain what recursion is and why it’s useful?**

**Recursion is a function that calls itself. It’s useful because it allows you to break down a problem into smaller, more manageable pieces.**