

List

Python offers a range of compounded data types often referred as sequences. List is one of the most frequently used and very versatile data types used.

How to create a list?

A list is created by placing all the items (elements) inside square brackets [], separated by commas.

It can have any number of items and they may be of different types (integer, float, string etc.).

List

Method

Description

append()

Adds an element at the end of the list

clear()

Removes all the elements from the list

copy()

Returns a copy of the list

count()

Returns the number of elements with specified value

extend()

Add the elements of a list to the end of the current list

index()

Return index of the first element with specified value

insert()

Adds an element at the specified position

pop()

Removes the element at the specified position

remove()

Removes the first item with the specified value

reverse()

Reverses the order of the list

sort()

Sorts the list

Python has a set of built-in methods that you can use on lists/arrays.

List- Example

List -Slicing

Tuples

A tuple in Python is similar to a list. The difference between the two is that we cannot change the elements of a tuple once it is assigned.

Creating a Tuple

A tuple is created by placing all the items (elements) inside parentheses (), separated by commas.

The parentheses are optional, but is a good practice to use

A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

Tuples

In Python, tuples are immutable. There are only two tuple methods count() and index() that a tuple object can call.

Python Tuple count()

returns count of the element in the tuple

Python Tuple index()

returns the index of the element in the tuple

Tuples

List vs Tuples

Exception Handling

Exceptions in Python

Python has many built-in exceptions that are raised when your program encounters an error (something in the program goes wrong). When these exceptions occur, the Python interpreter stops the current process and passes it to the calling process until it is handled. If not handled, the program will crash.

Example- Let us consider a program where we have a function A that calls function B, which in turn calls function C. If an exception occurs in function C but is not handled in C, the exception passes to B and then to A. If not handled, an error message is displayed and our program comes to a sudden unexpected halt.

Exception Handling

Exceptions can be raised for multiple reasons.

ImportError - An import fails
IndexError - A list is indexed without of range error
SyntaxError - Some syntax error
TypeError - A function is called with a wrong type of variable
ValueError - A function is called with inappropriate value

Exception Handling

Quiz

Q1. What is the difference between list and tuples in Python?

Q2. What is the purpose of the is not operator?

Q3. What is the role of range() function?

Q4. What are Import Error?

Q5. What is Exception Handling? How it is handled?

Thank You