Call by value and Call by reference

Call by value

(i) while calling a function, we pass the value of variables ~~to~~ to it. Such functions are known ~~as~~ as call by value

(ii) In this method, the value of each variable in the calling function is copied into corresponding dummy variables of the called function

(II) , with this method, the changes made to the dummy variables in the called function have no effect on the values of actual variable in the calling function

(iv) In call by value we cannot ~~a~~ alter the values of actual variables through function calls.

(v) values of variable are passed by the simple technique.

(vi) This method is preferred we have to pass some small values that should not change.

(VII) Call by value is considered safer as original data is preserved

call by reference

while calling a function, instead of passing the value of variables, we pass the address of variables (location of variables) to the function known as call by reference.

In this method, the address of actual variables in the calling function is copied into the dummy variables of the called function.

(III) with this method using addresses we would have access the actual variable and hence we would be able to manipulate them.

In call by reference we can alter the value of variables through function calls.

Pointer variables are necessary to define to store the address value of variables

This method is preferred when we have to pass a large amount of data to the function.

call by reference is risky as it allows direct ~~not~~ modification in original data

② Python Membership Operator

The python membership operator test for the membership of an object in a sequence, such as strings, lists or tuples, python offers two membership operators to check or validate the membership of a value. They are as follow

(i) Python IN operator

(ii) Python OUT operator

Python IN operator

The in operator is used to check if a character/substring/element exists in a sequence or not. Evaluate to true if it finds the specified element in a sequence otherwise false.

Python NOT in operator:-

The 'not' in python operator evaluates to true if it does not find the variables in the specified sequence and false otherwise.

Python Identity Operator

The Python identity Operators are used to Compare the objects if both the objects are actually of the same data type and share the same memory location. There are different identity Operator such as:

Python Is operator:-

This is operator evaluates to true if the variables on either side of the operator point to the same object in the memory and false otherwise.

Python IS NOT operator:

This is not operator value evaluates true if both variables on the either side of the operator are not the same object in the memory location Otherwise it evaluates false.

ASCII and UTF

ASCII and UTF-8 are character encoding standards used in python to represent text. ASCII uses 7 bits to represent 128 character, including uppercase and lowercase English letters numbers, punctuation, and Control characters. UTF-8 is a variable width encoding that can represent any Unicode character, using 1 to 4 bytes per character. It is a backward Compatible with ASCII. meaning that ASCII. Characters are represented using the same byte values in both encodings.