**Variable is a name that refers to a value its points to the memory location where data is read and modified.**

**Identifier is the name of a variable.**

**Constants are the fixed values, its also called literal: String, numeric, boolen, special(null), literal collections (list[], tuple(), set{},frozenset({}), dict{:} )**

**Data types: int, float, complex, bool, string, list , tuple, set, dict**

**Type() function: returns the type of given object**

**Format() function: returns a formatted representation of a given value.**

**Len() function: length of the data type**

**Id() : id of an object**

**Math modules:**

**abs(x)🡪returns distance b/w x and 0**

**round(x,n)🡪returns x rounded to n digits from the decimal point, ex:round(12345.9876,2)=12345.99**

**math.ceil(x)🡪it returns the integer(whole number) greater then x**

**math.floor(x)🡪it returns the integer(whole number) less then x**

**math.fabs(x)🡪it returns absolute value of x , ex: math.fabs(1)=1.0, math.fabs(5)=5.0**

**math.factorial(x)🡪returns x factorial, ex: math.factorial(4)=24**

**math.fsum(iterable)🡪it returns accurate floating point sum of values in the iterable, ex math.fsum([1, 2, 3, 4, 5])=15.0**

**math.pow(x,y)🡪returns value of x to the power of y, ex: math.pow(2,3)=8.0**

**math.sqrt(x)🡪return the square root of x for x greater then 0 ex: math.sqrt(9)=3.0**

**Random module:**

**random.random()🡪returns a random float in the interval [0.0,1.0]**

**random.randint(x,y)🡪generate a random integer b/w and including x and y**

**random.randrange(x,y)🡪generate number b/w x and y excluding y**

**random.uniform(x,y)🡪 generate random float number b/w and including x and y**

**random.choice(sequence)🡪 pick a random element from the sequence**

**random.shuffle(sequence)🡪 modify the sequence of the object**

**random.sample(sequence,n)🡪 pick a random sample of n unique elements**

**Operators:**

**Arithmetic operator🡪 (),Exponential(\*\*),modulus(%), floordivision(//),\*,+,\_ (PEMDAS)**

**Comparison operator🡪==, !=, <, >, <=, >=**

**Assignment operator🡪=, +=, -=,\*=, /=, %=, \*\*=, //=**

**Logical operator(Boolean Operators)🡪and, or, not**

**Membership operator🡪in, not in**

**Identity operator(compares the address of the object)🡪is, is not**

**Bitwise operator**

**& 🡪 Bitwise AND (if one value 0 then result is 0)**

**| 🡪 Bitwise OR (if one value 1 then result is 1)**

**^ 🡪 Bitwise Exclusive OR/ EXOR(if two values are same then result is 0)**

**~ 🡪 Bitwise complement**

**<< 🡪Shift left**

**>> 🡪Shift right**

**Int, float, complex, bool, str, tuple are immutable objects.**

**List, set, dict objects are mutable objects.**

**List is a collection which is ordered and changeable. Allows duplicate members.[ ]**

**Tuple is a collection which is ordered and unchangeable. Allows duplicate members.( )**

**Set is a collection which is unordered and unindexed. No duplicate members.{ }**

**Frozen Set🡪immutable form of set({})**

**Dictionary is a collection which is unordered, changeable and indexed. No duplicate members.{ }**

**Key is mutable, but value is immutable.**

**Print(Format(1234, ‘f’)) #1234.000000 🡪change the format of the element.**

**# String: (str)**

**Positive indexing starts from 0, Negative indexing start from -1**

**print(type(x)): returns the type of given object**

**print(len(x)): length of the data type**

**print(Id(x)) : id of an object**

**print(max(x)): Maximum value of the object**

**print(min(x)): minimum value of the object**

**Print(str1+str2)🡪concatenate to strings**

**str1= “How are you”**

**Print(str1.replace(“are”, “r”,1))🡪replace the characters in a string**

**Indexing🡪print(pystr[5])**

**Slicing🡪print(pystr[0:5])**

**Unicode string🡪string of any language can be stored in a variable i.e pystr= u “कुछ कुछ”**

**Capitalize🡪pystr.capitalize()🡪return the copy of string with 1st letter capital in a string**

**Title🡪pystr.title()🡪returns the copy of string with 1st letter of all the words in a string**

**Print(pystr.upper())🡪capitalize the entire string**

**Print(pystr.lower())🡪entire string into lower case**

**pystr.isdigit()🡪check if string consists of any digit.**

**pystr.islower()🡪Check if all characters are in lower case.**

**pystr.join()🡪in between the elements.**

**pystr=Python, Len=10**

**pystr.zfill(len)🡪it will display the string and empty space is filled with 0**

**pystr.rjust(len)🡪it will display the string with empty space in right**

**pystr.ljust(len)🡪 it will display the string with empty space in left**

**pystr.center(len)🡪 it will display the string with empty space in left and right**

**# list:**

**--------**

**1.concatination: print(list1+list2)**

**2.Repetition: print (list1\*2)**

**3.Membership: print(1 in list1)**

**4.length: print(len(list1))**

**5.index: print(list1[0])**

**6.slice:print(list1[1:4:2])**

**7.append: print( list1.append(3))**

**8.extend: print (list1.extend([2,3]))**

**9.insert: list1.insert(0,100)**

**10.remove: list1.remove(2)**

**11.clear: list1.clear()**

**12.index: list1.index("2")**

**13.count: list1.count(3)**

**14.sort: list1.sort()**

**15.reverse: list1.reverse()**

**16.pop(): list.pop()**

**17.max: max(list1)**

**18.min: min(list1)**

**19.sum: sum(list1)**

**# set:**

**------**

**1.add: set1.add(1)**

**2.clear: set1.clear()**

**3.update: set1.update(set2)**

**4.pop: set1.pop()**

**5.remove: set1.remove(2)**

**6.set1.difference(set2)🡪 remove all the elements of set2 from the set1**

**7.set1.difference\_update(set2)**

**8.set1.symmetric\_difference(set2)🡪 show the uncommon elements**

**9.set1.symmetric\_difference\_update(set2)**

**10.set1.issuperset(set2)**

**11.set1.issubset(set2)**

**12.set1.isdisjoint(set2)---> true if there is no common elements between them**

**#dictionary:**

**------------**

**1. pydict.clear()**

**2. pydict.copy()**

**3. pydict.get(1)---> gives value of key-1**

**4. pydict.items()**

**5. pydict.update(pydict2)**

**6. pydict.keys()--> display keys**

**7. pydict.values()--> display values**

**8. pydict.pop('key1')**

**9. pydict.popitem()**

**10. all() 🡪**

**11. any()**

**12. len()**

**13. sorted()**

**14. zlist=zip(pylist1,pylist2)..**

**Python Control Structure**

**Sequence Control Structure🡪execute line of code in sequential order**

**Selection Control Structure(conditional Control Structure)🡪 if, if..else, if ..elif..else, Nested if..else, Negative Conditions(if not condition)**

**Iterative Control Structure🡪for..else, while, nested loops, break, continue**

**For loop generally used with range function.**

**Range Function: It generates lists containing arithmetic operation**

**Range(stop) : print (list(range(10)))#[0,....,9]**

**Range(start, stop) : print(list(range(1,5)))#[1,...,4]**

**Range(start, stop, step) : print(list(range(1,10,2)))#[1,3,5,7.9]**

**Decrementing : (start value must be max)**

**print(list(range(10,2,-2)))#[10,8,6,4,]**

**Reading data from keyboad**

**x= int(input(“Enter Any Number :”))**