Strings

String is data type that stores a sequence of characters.

Basic Operations

- **concatenation**

  “hello” + “world” ⟶ “helloworld”

- **length of str**

  len(str)

Indexing

A p n a _ C o l l e g e

0 1 2 3 4 5 6 7 8 9 10 11

str = “Apna_College”

str[0] is ‘A’, str[1] is ‘p’ ...

str[0] = ‘B’ #not allowed

Slicing

Accessing parts of a string

```
str[ starting_idx : ending_idx ] #ending idx is not included
```

str = “ApnaCollege”

str[1 : 4] is “pna”

str[: 4] is same as str[0 : 4]

str[1 :] is same as str[1 : len(str)]

Slicing

Negative Index

A p p l e

-5 -4 -3 -2 -1

str = “Apple”

str[-3 : -1] is “pl”

String Functions

```
str = "I am a coder."

str.endsWith("er.")  #returns true if string ends with substr

str.capitalize( )  #capitalizes 1st char

str.replace( old, new )  #replaces all occurrences of old with new

str.find( word )  #returns 1st index of 1st occurrence

str.count("am")  #counts the occurrence of substr in string
```

Let‘s Practice

WAP to input user’s first name & print its length.

WAP to find the occurrence of ‘$’ in a String.

Conditional Statements

if-elif-else (SYNTAX)

```
if(condition) :
        Statement1
elif(condition):
        Statement2
else:
        StatementN
```

Conditional Statements

Grade students based on marks

marks >= 90, grade = “A”

90 > marks >= 80, grade = “B”

80 > marks >= 70, grade = “C”

70 > marks, grade = “D”

Let‘s Practice

WAP to check if a number entered by the user is odd or even.

WAP to find the greatest of 3 numbers entered by the user.

WAP to check if a number is a multiple of 7 or not.

Lists in Python

A built-in data type that stores set of values

It can store elements of different types (integer, float, string, etc.)


```
marks = [87, 64, 33, 95, 76]          #marks[0], marks[1]..

student = [”Karan”, 85, “Delhi”]      #student[0], student[1]..

student[0] = “Arjun”      #allowed in python

len(student)      #returns length
```

List Slicing

Similar to String Slicing

```
list_name[ starting_idx : ending_idx ] #ending idx is not included
```

marks = [87, 64, 33, 95, 76]

marks[1 : 4] is [64, 33, 95]

marks[: 4] is same as marks[0 : 4]

marks[1 :] is same as marks[1 : len(marks)]

marks[-3 : -1] is [33, 95]

List Methods

```
list = [2, 1, 3]

list.append(4)  #adds one element at the end      [2, 1, 3, 4]

list.sort( )  #sorts in ascending order           [1, 2, 3]

list.sort( reverse=True )  #sorts in descending order    [3, 2, 1]

list.reverse( )  #reverses list          [3, 1, 2]

list.insert( idx, el )  #insert element at index
```

List Methods

```
list = [2, 1, 3, 1]

list.remove(1)  #removes first occurrence of element    [2, 3, 1]

list.pop( idx )  #removes element at idx
```

Tuples in Python

A built-in data type that lets us create immutable sequences of values.

```
tup = (87, 64, 33, 95, 76)    #tup[0], tup[1]..

tup[0] = 43      #NOT allowed in python


tup1 = ( )

tup2 = ( 1, )

tup3 = ( 1, 2, 3 )
```

Tuple Methods

```
tup = (2, 1, 3, 1)

tup.index( el )  #returns index of first occurrence    tup.index(1) is 1

tup.count( el )  #counts total occurrences      tup.count(1) is 2
```

Let‘s Practice

WAP to ask the user to enter names of their 3 favorite movies & store them in a list.

WAP to check if a list contains a palindrome of elements. (Hint: use copy( ) method)

[1, 2, 3, 2, 1]

[1, “abc”, “abc”, 1]

Let‘s Practice

WAP to count the number of students with the “A” grade in the following tuple.

[”C”, “D”, “A”, “A”, “B”, “B”, “A”]

Store the above values in a list & sort them from “A” to “D”.