Python

String:

```
#strig basic
n=input()
print(n)
for i in n:
  print(i,end="")
print()
#present in string or not
if "r" in n:
  print("Yes r present")
else:
  print("NO")
if "k" not in n:
  print("Yes k is not here")
else:
  print("Opps k is here")

#string reverse
def reverse(n):
  return n[::-1]
print("Reverse: "+reverse(n))

#Slicing Strings
"""Get the characters from position 2 to position 5"""
k=("Hello, world!")
```

```
print(k[2:5])

"""Get the characters from the start to position 5 """

print(k[:5])

"""Get the characters from position 2, and all the way to the end"""

print(k[2:])

"""Get the characters: From: "o" in "World!" (position -5) To, but not included: "d" in "World!" (position -2)"""

print(k[-5:-2])


#Uppercase

print(k.upper())

#lowercase

print(k.lower())

#Strip()-Removes space from the beginning or the end

k=" Hello, World!"

print(k.strip())

#replace() method replaces a string with another string

k="Hello, World!"

print(k.replace("H","J"))

#split() method splits the string into substrings if it finds instances of the separator

print(k.split())

#format() method to insert numbers into strings

age =22

k="My age is {}"

print(k.format(age))
```

```
result=3.48

income=0

k="My name is pronoy. My age is {}. My cgpa is {}. My monthly
income is {}."

print(k.format(age,result,income))

k="My name is pronoy. My age is {2}. My cgpa is {0}. My monthly
income is {1}."

print(k.format(result,income,age))

k="My name is \"Pronoy\""

print(k)
```

List:

```
a=['a','b','c']

print(a)

print(a[0])#index

print(a[-1])#last element

print(a[-2])#2nd last element

a=["apple", "banana", "cherry", "orange", "kiwi", "melon",
"mango"]

print(a[2:5])#3rd to 5th element

print(a[2:])#after 2nd elemt

print(a[:4])#firth to 4th index

print(a[-4:-1])#last 4 th elemet to 2nd last element

if "apple" in a:#item exists
```

```
    print("Yes apple exits")
else:
    print("sorry")
a[1]="pinaple"#change item value
print(a)
a[1:3]=["pinaple","watermelon"]
print(a)
#If you insert more items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly:
a= ["apple", "banana", "cherry"]
a[1:2]=["blackcurrant", "watermelon"]
print(a)
#insert()-inserts an item at the specified index
a= ["apple", "banana", "cherry"]
a.insert(2,"Watermelom")
print(a)
#append() method to append an item
a= ["apple", "banana", "cherry"]
a.append("orange")
print(a)
#To append elements from another list to the current list, use the extend() method
a= ["apple", "banana", "cherry"]
b= ["mango", "pineapple", "papaya"]
a.extend(b)
print(a)
```

```
#The extend() method does not have to append lists, you can add any iterable object (tuples, sets, dictionaries etc.).
a= ["apple", "banana", "cherry"]
b= ("mango", "pineapple", "papaya")
a.extend(b)
#The remove() method removes the specified item
a= ["apple", "banana", "cherry"]
a.remove("banana")
print(a)
#The pop() method removes the specified index
"""If you do not specify the index, the pop() method removes the last item"""
a= ["apple", "banana", "cherry"]
a.pop(1)
print(a)
#The del keyword also removes the specified index
a= ["apple", "banana", "cherry"]
del a[0]
print(a)
"""The del keyword can also delete the list completely."""
a= ["apple", "banana", "cherry"]
del a
#clear() method empties the list
a= ["apple", "banana", "cherry"]
a.clear()
print(a)
```

```python
#Loop in list
#Print all items in the list, one by one
a= ["apple", "banana", "cherry"]
for i in a:
    print(i)
#rint all items by referring to their index number
for i in range(len(a)):
    print(a[i])

i=0
while i<len(a):
    print(a[i])
    i+=1

new=[]
for i in a:
    if 'a' in i:
        new.append(i)
print(new)

new=[]
for i in a:
    new.append(i.upper())
print(new)

#sort list
```

```
a = ["orange", "mango", "kiwi", "pineapple", "banana"]
b=sorted(a)
print(b)

#sort desending
b=sorted(a,reverse=True)
print(b)
```

Tuples:

```
t=("apple", "banana", "cherry")
print(t)
#tupple length
print(len(t))
#Print the second item in the tuple
print(t[1])
#Print the last item of the tuple
print(t[-1])
#Return the third, fourth, and fifth item
t=("apple", "banana", "cherry", "orange", "kiwi", "melon",
"mango")
print(t[2:5])
"""The search will start at index 2 (included) and end at index
5 (not included).
    the first index has 0"""
print(t[:4])
```

```
print(t[2:])

print(t[-4:-1])

#Check if "apple" is present in the tuple

if "apple" in t:

    print("YES")

"""Once a tuple is created, you cannot change its values.
Tuples are unchangeable, or immutable as it also is called.

   But there is a workaround. You can convert the tuple into a
list, change the list, and convert the list back into a
tuple."""

#Convert the tuple into a list to be able to change it

x=list(t)

x[1]="kiwi"

t=tuple(x)

print(t)

#Unpacking a Tuple

"""When we create a tuple, we normally assign values to it.
This is called "packing" a tuple

    But, in Python, we are also allowed to extract the values
back into variables. This is called unpacking"""

t=("apple", "banana", "cherry")

x,y,z=t

print(x)

#and all of this is like list
```

Set:

```
"""Set items are unchangeable, but you can remove items and add
new items.
and do not allow duplicate values."""
s={"apple", "banana", "cherry"}
print(s)
#length of a set
print(len(s))
#Duplicate values will be ignored
s = {"apple", "banana", "cherry", "apple"}
print(s)
#Loop through the set, and print the values
for i in s:
    print(i)
if "banana" in s:
    print("YES")
#Add Items
s = {"apple", "banana", "cherry"}
s.add("orange")
print(s)
#To add items from another set into the current set, use the
update() method.
s = {"apple", "banana", "cherry"}
s1={"pineapple", "mango", "papaya"}
s.update(s1)
print(s)
```

```
"""The object in the update() method does not have to be a set,
it can be any iterable object (tuples, lists, dictionaries
etc.)."""

#To remove an item in a set, use the remove(), or the discard()
method.

s = {"apple", "banana", "cherry"}

s.remove("banana")

print(s)

"""Note: If the item to remove does not exist, remove() will
raise an error."""

s = {"apple", "banana", "cherry"}

s.discard("banana")

print(s)

"""Note: If the item to remove does not exist, discard() will
NOT raise an error."""

#Remove a random item by using the pop() method

s = {"apple", "banana", "cherry"}

s.pop()

print(s)

"""Sets are unordered, so when using the pop() method, you do
not know which item that gets removed."""

#The clear() method empties the set

s = {"apple", "banana", "cherry"}

s.clear()

print(s)

#The del keyword will delete the set completely

s = {"apple", "banana", "cherry"}

del s
```

```
#Loop through the set, and print the values
s = {"apple", "banana", "cherry"}
for i in s:
    print(i)
#You can use the union() method that returns a new set
containing all items from both sets, or the update() method
that inserts all the items from one set into another
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}


set3 = set1.union(set2)
print(set3)


#Keep ONLY the Duplicates
"""The intersection_update() method will keep only the items
that are present in both sets."""
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
x.intersection_update(y)
print(x)
"""The intersection() method will return a new set, that only
contains the items that are present in both sets."""
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
z = x.intersection(y)
print(z)
```