In [8]:

```
# to check the given character is vowels or constant
#vowels;a,e,i,o,u
ch=string(input("enter character..")) ch=i
if(ch=-'a'or ch=='e'or ch=='i' or ch=='o' or ch =='u'):
      print(ch,"it is owel")
        else:
    print(charachter,"it is constnt")
```

```
  File "<tokenize>", line 7
    print(charachter,"it is constnt")
    ^
IndentationError: unindent does not match any outer indentation level
```

In [14]:

```
<p style="color":green">prasanna</p>
```

```
  File "<ipython-input-14-3174b2781eb6>", line 1
    <p style="color":green">prasanna</p>
    ^
SyntaxError: invalid syntax
```

In [ ]:

```


n1=int(input("enter n1 value..."))
n2=int(input("enter n2 value..."))
n3=int(input("enter n3 value..."))
if(n1==n2 and n2==n3):
    print("is three are equal..")
    elif(n1>n2a and n1>n3):
    print ("n1 is grater then n1 and n3")
    elif(n2>n3):
```

In [ ]:

```
<img src="
```

In [5]:

```
# to print the 1 to 10 natural numbers by using for loop
for i in  range (11):
    print(i, end ="")
```

```
012345678910
```

In [10]:

```
# to print the odd numbers from s 1 eto 100 by using for loop
for i in range(1,100,2):
  print(i,end="")
```

```
13579111315171921232527293133353739414345474951535557596163656769717375777981838587899193959799
```

In [12]:

```
#lit 3 eliments to print the value starting character 0 ending char 50 to sp
for i in range (0, 50, 3):
      print(i, end="")
```

```
0369121518212427303336394245 48
```

In [23]:

```
# to print the natural numbers in assing order
n=int(input("enter a natural number size"))
for i in range (1,n+1):
      print(i,end="")
```

```
enter a natural number size20
1234567891011121314151617181920
```

In [46]:

```
for i in range(4):
    if i==4:
        break
    else:
        print(i,end=" ")
```

```
0 1 2 3
```

In [51]:

```
# to print  the range of 1 to
for i in range(1,10):
    if i==5:
        break
    else:
            print(i,end="")
```

```
1234
```

In [53]:

```
# to print
for i range(1,41):
    iF(i%2!=0):
        continue
        else:
            print(i,end=" ")
```

```
  File "<ipython-input-53-1ce0c62c138f>", line 2
    for i range(1,41):
          ^
SyntaxError: invalid syntax
```

In [ ]:

```
# swaqp b/w two numbers
a=5
b=10
input:5,10
    output=10,5
```

In [54]:

```
a=5
b=15
print("before swaping",a)
print("before swaping",b)
temp=a
a=b
#b=5
b=temp
#b=15
print("after swaping:a value",a)
print
```

```
  File "<ipython-input-54-01a061a0bbf4>", line 4
    print("value of a  :"a, " and b:",b)
                         ^
SyntaxError: invalid syntax
```

In [57]:

```

# how to generate a random number
#between 0 and 99

import random

print(random.randint(0,99))
```

```
51
```

In [58]:

```
#to print the alphabeta
import string
print("Alphabet from a-z:")
for letter in string.ascii_lowercase:
   print(letter, end =" ")
print("\nAlphabet from A-Z:")
for letter in string.ascii_uppercase:
   print(letter, end =" ")
```

```
Alphabet from a-z:
a b c d e f g h i j k l m n o p q r s t u v w x y z
Alphabet from A-Z:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

In [69]:

```
# to program to displey calender of the given month and year
import calendar
print(calendar.month(2022,11))
```

```
   November 2022
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

In [71]:

```
# program to disply calender of the current year and month
import.calendar
year=2022
month=11
print(calande.month(year))
```

```
  File "<ipython-input-71-ba83b313044b>", line 2
    import.calendar
           ^
SyntaxError: invalid syntax
```

In [73]:

```
import calendar
year=2012
month=11
print(calendar.month(year,month))
```

```
   November 2012
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

In [76]:

```
def function_name(argument_list):
    statements
    retuen value
```

```
  File "<ipython-input-76-6fadff09fd2e>", line 3
    retuen value
           ^
SyntaxError: invalid syntax
```

In [1]:

```
a=int(input("enter a value"))
b=int(input("entern b value"))
def add(a,b):
    c=a+b
return c
add(a,b)
```

```
enter a value10
entern b value10
```

```
  File "<ipython-input-1-79a4d87b1958>", line 5
    return c
    ^
SyntaxError: 'return' outside function
```

In [2]:

```
a=int(input("enter a value"))
b=int(input("enter b value"))
def add(a,b):
    c=a+b
    return c
add(a,b)
```

```
enter a value12
enter b value12
```

Out[2]:

```
24
```

In [4]:

```
n1=20
n2=10
def sub(n1,n2):
    c=n1-n2
    return c
sub(n1,n2)
```

Out[4]:

```
10
```

In [ ]:

```
n1=25
n2=10
def add(n1,n2)
```

In [5]:

```
a=int(input("enter a value"))
```

```
enter a value65
```

In [6]:

```
def my_function(x):
  return 5 * x

print(my_function(3))
print(my_function(5))
print(my_function(9))
```

```
15
25
45
```

In [9]:

```
def add():
    a=23
    b=26
    add=a+b
    return add
add()
```

Out[9]:

```
49
```

In [12]:

```
def my_func1():
  print (def my_func1():
  print ("Hello World")
  return None

def my_func2():
  print ("Hello World")
  return

def my_func3():
  print("Hello World")
  return None

def my_func2():
  print("Hello World")
  return

def my_func3():
  print ("Hello World")
```

```
  File "<ipython-input-12-d2b4735b7a21>", line 2
    print (def my_func1():
           ^
SyntaxError: invalid syntax
```

In [1]:

```
from trurtle import*
color("read")
begin_fill()
pensize(3)
left(50)
forward(133)
cricle(50,200)
right(140)
circle(50,200)
forward(133)
end_fill()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-946f57afe084> in <module>
----> 1 from trurtle import*
      2 color("read")
      3 begin_fill()
      4 pensize(3)
      5 left(50)

ModuleNotFoundError: No module named 'trurtle'
```

In [ ]:

```
# a list ifs a collection of characters variables, and
# number variables and booleam values datatypes
# a is a store multiple data with a single variable
# sa list is a orderd type of data
# a list is denoted as []
#a lis is iteam as denoted with dobule quotes.

syntax:
    items=["iteam","iteam2", "iteam3"]
    print(iteams)
```

In [6]:

```
# exaple for the list
li=["apple", "orange", "bananna", "grape"]
print(li)
```

```
['apple', 'orange', 'bananna', 'grape']
```

In [7]:

```
# type of the list
print(type(li))
```

```
<class 'list'>
```

In [ ]:

```

```

In [ ]:

```
```

In [9]:

```
# length if the list
print(len(li))
```

```
4
```

In [21]:

```
# accicing the item in list of not
if "apple" in li:
    print("yes")
    else:
        print("no")
```

```
  File "<ipython-input-21-f3196ed74535>", line 4
    else:
    ^
SyntaxError: invalid syntax
```

In [ ]:

```
li[0]=p
```

In [25]:

```
li.insert(1,"prasanna")
li
```

Out[25]:

```
['apple', 'prasanna', 'prasanna', 'orange', 'bananna', 'grape']
```

In [16]:

```
li
```

Out[16]:

```
['apple', 'orange', 'bananna', 'grape']
```

In [18]:

```
li[2:4]
```

Out[18]:

```
['bananna', 'grape']
```

In [19]:

```
li[2:]
```

Out[19]:

```
['bananna', 'grape']
```

In [20]:

```
li[:3]
```

Out[20]:

```
['apple', 'orange', 'bananna']
```

In [32]:

```
li.remove("prasanna")
```

In [34]:

```
li
```

Out[34]:

```
['orange', 'bananna', 'grape']
```

In [37]:

```
li1=["anusha", "madhu" ,"chaitu"]
print(li)
```

```
['orange', 'bananna', 'grape']
```

In [39]:

```
del li[1]
```

In [40]:

```
li
```

Out[40]:

```
['orange']
```

In [42]:

```
# list using loop
for i in li:
    print(i, end)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-42-5c4556298896> in <module>
      1 # list using loop
      2 for i in li:
----> 3     print(i, end)

NameError: name 'end' is not defined
```

In [43]:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

```
('apple', 'banana', 'cherry')
```

In [44]:

```
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
```

```
('apple', 'banana', 'cherry', 'apple', 'cherry')
```

In [45]:

```
thistuple = ("apple",)
print(type(thistuple))

#NOT a tuple
thistuple = ("apple")
print(type(thistuple))
```

```
<class 'tuple'>
<class 'str'>
```

In [58]:

```
# String, int and boolean data types:
```

In [57]:

```
tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = (True, False, False)

print(tuple1)
print(tuple2)
print(tuple3)
```

```
('apple', 'banana', 'cherry')
(1, 5, 7, 9, 3)
(True, False, False)
```

tuple

it is a collection of different types of data.

it is immutable (can't changed)

we can use round brackets() to write a tuple.

to create the empty tuple

tuple_name=()

to create single values

tuple_name=(values)

to create multiple values

tuple_name=(values1, value2,..)

In [63]:

```
#create tuple
t1=(10,20,30)
t1
print(type(t1))
```

```
<class 'tuple'>
```

In [65]:

```
# single value tuple
t2=(10)
print(type(t2))
t3=(20,)
print(type(t3))
```

```
<class 'int'>
<class 'tuple'>
```

In [66]:

```
t3
```

Out[66]:

```
(20,)
```

In [67]:

```
# ti create multiple value
t1
print(t1[2])
```

```
30
```

In [69]:

```
print(t1[0:1])
```

```
(10,)
```

In [78]:

```
t2=(10,20,40,50,60,70,40,40)
    # to count the number if ocurences
t2.count(10)
```

Out[78]:

```
1
```

In [79]:

```
#index
t2.index(40)
```

Out[79]:

```
2
```

In [2]:

```
tuple1 =("abc", 34, true, 40, "male")
print(tuple1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-153f725bdc2e> in <module>
----> 1 tuple1 =("abc", 34, true, 40, "male")
      2 print(tuple1)

NameError: name 'true' is not defined
```

In [ ]:

```
#dictionary:
-it is collection of diffrent data types.
- it is group of key woards
```

In [14]:

```
d1={'a':10,'b':34,'c':45}
print(d1)
print(type(d1))
```

```
{'a': 10, 'b': 34, 'c': 45}
<class 'dict'>
```

In [15]:

```
# to create a dictionaries with diffrent data types..
d2={'a':100, 'name':'anusha','branch': 'mba', 'b':45.8}
print(d2)
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'b': 45.8}
```

In [16]:

```
print(d2)
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'b': 45.8}
```

In [ ]:

```
#copy
printI(d2)
d3=d2.copy()
print(d3)
```

In [24]:

```
#get
print(d2)
print(d2.get('a'))
print(d2.get('name'))
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba'}
100
anusha
```

In [25]:

```
#setdefault
print(d2)
print(d2.setdefault('rollno',310))
print(d2)
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba'}
310
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'rollno': 310}
```

In [ ]:

```

```

In [17]:

```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__do
c__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__
repr__', '__reversed__', '__setattr__', '__setitem__', '__sizeof__', '__str_
_', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys',
'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [20]:

```
#keys
print(d2)
print(d2.keys())
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'b': 45.8}
dict_keys(['a', 'name', 'branch', 'b'])
```

In [21]:

```
#pop
print(d2)
print(d2.pop('b'))
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'b': 45.8}
45.8
```

In [32]:

```
#popitem
print(d2)
print(d2.popitem())
```

```
{}
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-32-3147d90ab6d3> in <module>
      1 #popitem
      2 print(d2)
----> 3 print(d2.popitem())

KeyError: 'popitem(): dictionary is empty'
```

In [ ]:

In [22]:

```
#values()
print(d2)
print(d2.values())
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba'}
dict_values([100, 'anusha', 'mba'])
```

In [29]:

```
#clear
print(d2)
print(d2.clear())
```

```
{'a': 100, 'name': 'anusha', 'branch': 'mba', 'rollno': 310}
None
```

In [ ]:

```

```

In [ ]:

```

```