

```
In [1]: # Criteria of best Prpgram
# 1> Syntax - way to write code
# 2> Semantic - meaning of program
# 3> Performance - time of execution
# 4> Understandability - it helps in modification or upadtion in program
```

INPUT

```
In [3]: i=input("Enter number: ")
print(i)
print(type(i))
```

```
Enter number: 45
45
<class 'str'>
```

```
In [4]: i=input("Enter number: ")
print(i)
print(type(i))
j=int(i)
print(j)
print(type(j))
```

```
Enter number: 45
45
<class 'str'>
45
<class 'int'>
```

```
In [5]: x=input("Enter number: ")
y=input("Enter number: ")
z=input("Enter number: ")
```

```
Enter number: 5
Enter number: 54
Enter number: 1
```

```
In [6]: print("Enter 3 no.:")
for i in range(0,3):
    input()
```

```
Enter 3 no.:
5
6
8
```

```
In [7]: x=input("Enter the no.:").split()
print(x)
print(type(x))
```

```
Enter the no.:5 4 8 9 6 5 2 3 6 4
['5', '4', '8', '9', '6', '5', '2', '3', '6', '4']
<class 'list'>
```

```
In [9]: x=int(input("Enter two no.:").split())
print(x)
print(type(x))
```

```
Enter two no.:88 2
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[9], line 1
----> 1 x=int(input("Enter two no.:").split())
      2 print(x)
      3 print(type(x))
```

TypeError: int() argument must be a string, a bytes-like object or a real number, not 'list'

```
In [11]: x=int(input("Enter two no.:"))
print(x)
print(type(x))
```

```
Enter two no.:5 6
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[11], line 1
----> 1 x=int(input("Enter two no.:"))
      2 print(x)
      3 print(type(x))
```

ValueError: invalid literal for int() with base 10: '5 6'

```
In [10]: x,y=input("Enter two no.:").split()
print(int(x),int(y))
print(type(x),type(y))
```

```
Enter two no.:54 65
54 65
<class 'str'> <class 'str'>
```

```
In [13]: x,y=[int(i) for i in input("Enter two no.:").split()]
print(x,y)
print(type(x),type(y))
print(x+y)
```

```
Enter two no.:5 6
5 6
<class 'int'> <class 'int'>
11
```

```
In [14]: x=[int(i) for i in input("Enter numbers with spaces:").split()]
print(x,y)
print(type(x))
print(sum(x))
```

```
Enter numbers with spaces:5 6 3 2 1 4 7 8 9 5
[5, 6, 3, 2, 1, 4, 7, 8, 9, 5] 6
<class 'list'>
50
```

```
In [16]: # Print average of a list by taking input
x=[int(i) for i in input("Enter numbers with spaces:").split()]
print(x)
print(type(x))
print(sum(x)/len(x))
```

```
Enter numbers with spaces:5 6 7 8 9 10 11 12
[5, 6, 7, 8, 9, 10, 11, 12]
<class 'list'>
8.5
```

```
In [17]: x=input("Enter text:")
print(x)
print(x.split())
```

```
Enter text:nikhil aids third sem 2023
nikhil aids third sem 2023
['nikhil', 'aids', 'third', 'sem', '2023']
```

OUTPUT

```
In [21]: # formating of string
name=input("enter name:")
age=int(input("enter age:"))
weight=float(input("enter weight:"))
print(" This is name: {}\n This is age: {} \n This is weight: {}".format(name,
# it put down the values in {} orderly
# format is f() it called by . operator
```

```
enter name:Nikhil
enter age:18
enter weight:49.26
This is name: Nikhil
This is age: 18
This is weight: 49.26
```

```
In [35]: name="Nikhil"
age=18
weight=49.26
print(" This is name: {}\n This is age: {} \n This is weight: {}".format(name,
print("#####")
print(" This is name: %s\n This is age: %d \n This is weight: %f"%(name,age,we
# %i --> int
# %d --> decimal
# %hx --> hexa decimal
# %o --> octal
# %s --> string
# %f --> float
```

```
This is name: Nikhil
This is age: 18
This is weight: 49.26
#####
This is name: Nikhil
This is age: 18
This is weight: 49.260000
```

```
In [39]: x=888888888.264
print("This is float value: {}".format(x))
print("This is float value: {:.f}".format(x))
print("This is float value: {:.3f}".format(x))
print("This is float value: {:.12.9f}".format(x))
print("This is float value: {:.2.15f}".format(x))
```

```
This is float value: 888888888.264
This is float value: 888888888.264000
This is float value: 888888888.264
This is float value: 888888888.264000058
This is float value: 888888888.264000058174133
```

```
In [40]: x=[54,69,42]
print("This is value: {}".format(x))
```

```
This is value: [54, 69, 42]
```

```
In [41]: y={1:"Nikhil",2:"Pankaj",3:"Ankush"}
print("This is value: {}".format(y))
```

This is value: {1: 'Nikhil', 2: 'Pankaj', 3: 'Ankush'}

```
In [43]: # F-string
name="Nikhil"
age=18
weight=49.26
print(f" This is name: {name}\n This is age: {age} \n This is weight: {weight}")
```

This is name: Nikhil
This is age: 18
This is weight: 49.26

```
In [44]: print(" This is name: {name}\n This is age: {age} \n This is weight: {weight}")
```

This is name: {name}
This is age: {age}
This is weight: {weight}

Function

```
In [45]: # 1st f() Definition

# def say_Hi(arguments):      # def is a keyword to define f() -> name of f() like
#     return value           # return is a keyword which is optional it gives a

# f() Calling

def function1():
    print("My 1st Function")    # f() doesn't execute on definition time
```

```
In [46]: function1()    # function calling
```

My 1st Function

```
In [47]: def add():
    a=int(input("enter 1st no.:"))
    b=int(input("enter 2nd no.:"))
    print("sum is:",a+b)
```

add()

enter 1st no.:12
enter 2nd no.:45
sum is: 57

```
In [50]: def add(a,b):
          print("sum is:",a+b)

          add()
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[50], line 4
      1 def add(a,b):
      2     print("sum is:",a+b)
----> 4 add()
```

TypeError: add() missing 2 required positional arguments: 'a' and 'b'

```
In [51]: def add(a,b):
          print("sum is:",a+b)

          add(4,9)
```

sum is: 13

```
In [52]: def add(a,b):
          print("sum is:",a+b)

          a=int(input("enter 1st no.:"))
          b=int(input("enter 2nd no.:"))
          add(a,b)
```

enter 1st no.:5
enter 2nd no.:6
sum is: 11

```
In [54]: def add(a,b):
          print("sum is:",a+b)  # print sum of previous values of a & b

          a=int(input("enter 1st no.:"))
          b=int(input("enter 2nd no.:"))
          print("inside f()")
          print("outside f()")
          add(a,b)
```

outside f()
sum is: 11
enter 1st no.:2
enter 2nd no.:4
inside f()

```
In [56]: def add(a,b):
          print("sum is:",a+b)

          a=int(input("enter 1st no.:"))
          b=int(input("enter 2nd no.:"))
          print("inside f()")
print("outside f()")
add(m,n) # throws error
```

outside f()

```
-----
NameError                                Traceback (most recent call last)
Cell In[56], line 8
      6     print("inside f()")
      7     print("outside f()")
----> 8     add(m,n)

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

```
In [58]: def add(a,b):
          print("sum is:",a+b)

          a=int(input("enter 1st no.:"))
          b=int(input("enter 2nd no.:"))
          print("inside f()")
print("outside f()")
m=9
n=8
add(m,n)
```

outside f()
sum is: 17
enter 1st no.:4
enter 2nd no.:7
inside f()

```
In [61]: def add(a,b):
          print("inside f()")
          return a+b

print("outside f()")
a=int(input("enter 1st no.:"))
b=int(input("enter 2nd no.:"))
x=add(a,b)
print("x :",x)
print("add(a,b) :",add(a,b))
```

outside f()
enter 1st no.:12
enter 2nd no.:13
inside f()
x : 25
inside f()
add(a,b) : 25

```
In [62]: def square(n):  
         return n**2  
         s=square(7)  
         print(s)
```

49

```
In [ ]:
```