Print name

```
print("vishnu")
```

```
vishnu
```

Create variables of various numeric data types and assigning them values.Print the data types and values of these variables.

```
a=5
b=6.5
c=2+4j
print(f"Type of variable a and its value :{type(a),a}")
print(f"Type of variable b and its value :{type(b),b}")
print(f"Type of variable c and its value :{type(c),c}")
```

```
Type of variable a and its value :(<class 'int'>, 5)
Type of variable b and its value :(<class 'float'>, 6.5)
Type of variable c and its value :(<class 'complex'>, (2+4j))
```

Perform mathematical operations on these variables.

```
sum=b+a
sum2=c+b
diff=b-a
mul=b*a
div=b/a
print("sum :",sum)
print("sum2 :",sum2)
print("diff :",diff)
print("mul :",mul)
print("div :",div)
```

```
sum : 11.5
sum2 : (8.5+4j)
diff : 1.5
mul : 32.5
div : 1.3
```

Update the values of these variables.

```
a = 10
print(type(a))
print(a)

b = 8.5
print(type(b))
print(b)

c = 3 + 6j
print(type(c))
print(c)
```

```
<class 'int'>
10
<class 'float'>
8.5
<class 'complex'>
(3+6j)
```

Create boolean variables with True or False values.Print the data types of these boolean variables

```
d = True
e = False
print(type(d))
print(type(e))

print("Logical AND:",d and e)
print("Logical OR:",d or e)
print("LOgical NOT:",not d)
```

```
<class 'bool'>
<class 'bool'>
Logical AND: False
Logical OR: True
LOgical NOT: False
```

Create string variables with text values.Print the contents and lengths of these string variables and Concatenate strings.

```
str1 = "   hello"
str2 = "WORLD  "
print(str1, len(str1))
print(str2,len(str2))

str3 = str1 +' '+ str2
print(str3)
```

```
   hello 8
WORLD   7
   hello WORLD
```

Format strings with variables.Use string methods to manipulate strings by capitalizing, converting to uppercase, justifying, centering, replacing substrings, and stripping whitespace.

```
name="vishnu"
age="23"
print("my name is {} and my age is {}".format(name,age))

print(str1.capitalize())
print(str1.upper())
print(str2.lower())
print(str3.ljust(10))
print(str3.center(10))
print(str3.replace("World", "Python"))
print(str3.strip())
```

```
my name is vishnu and my age is 23
   hello
   HELLO
world
   hello WORLD
   hello WORLD
   hello WORLD
hello WORLD
```

Create and use Python lists. Perform tasks like appending elements, indexing, slicing, and iterating through the list.

```
list1=[1,2, 'fig',3.5, True]
list1.append(6)
print(list1)
print(list1[2])
print(list1[1:4])
for i,ele in enumerate(list1):
  print('#{}:{}'.format(i+1,ele))
```

```
[1, 2, 'fig', 3.5, True, 6]
fig
[2, 'fig', 3.5]
#1:1
#2:2
#3:fig
#4:3.5
#5:True
#6:6
```

Create and use Python tuples. Perform tasks like indexing, slicing, and concatenation.

```
tuple1 = (1, 2, 3, 4, 5)
print(tuple1[0])
print(tuple1[2:])
tuple2=(6,7)
print(tuple1 + tuple2)
```

```
1
(3, 4, 5)
(1, 2, 3, 4, 5, 6, 7)
```

Create and use Python sets. Perform tasks like accessing, adding, deleting set elements.

```
set1 = {1,2,3,5,6}
print(set1)
set1.add(7)
print(set1)
set1.remove(2)
print(set1)
```

```
{1, 2, 3, 5, 6}
{1, 2, 3, 5, 6, 7}
{1, 3, 5, 6, 7}
```

Create and use Python dictionaries. Perform tasks like adding, updating, and removing key-value pairs, and accessing values.

```
dict1 = {'name': 'vishnu', 'age': 23}
print(dict1)
dict1['city'] = 'New York'
print(dict1)
dict1['age'] = 24
print(dict1)
del dict1['age']
print(dict1)
print(dict1['name'])
```

```
{'name': 'vishnu', 'age': 23}
{'name': 'vishnu', 'age': 23, 'city': 'New York'}
{'name': 'vishnu', 'age': 24, 'city': 'New York'}
{'name': 'vishnu', 'city': 'New York'}
vishnu
```

Define simple functions with parameters and return values.

```
def sign(x):
    if x > 0:
        return 'positive'
    elif x < 0:
        return 'negative'
    else:
        return 'zero'

for x in [-1, 0, 1]:
    print(sign(x))
```

```
negative
zero
positive
```

Call functions with different arguments and use the returned results.Write functions that accept other functions as arguments.

```
def apply(func, x):
  return func(x)

def square(x):
  return x * x
result = apply(square, a)
print(result)
```

```
25
```

Define and use Python classes. Include tasks like creating a class, defining methods, and creating instances.

```
class car:
  def __init__(self, brand, model):
    self.brand = brand
    self.model = model
  def display_info(self):
    print(f"This is a {self.brand} {self.model}")
my_car = car("Toyota", "Camry")
my_car.display_info()
```

```
This is a Toyota Camry
```

Implement class inheritance and method overriding.

```
class seater(car):
  def __init__(self, brand, model, seats):
    super().__init__(brand, model)
    self.seats = seats
  def display_info(self):
    super().display_info()
    print(f"It has {self.seats} seats")
my_seater = seater("Toyota", "Camry", 5)
my_seater.display_info()
```

```
This is a Toyota Camry
It has 5 seats
```

Create a class with class variables and instance variables, and demonstrate their usage.

```
class Vehicle:
  vehicle_type = "Car"

  def __init__(self, brand, model):
    self.brand = brand
    self.model = model

  def display_info(self):
    print(f"Vehicle Type: {Vehicle.vehicle_type}")
    print(f"Brand: {self.brand}")
    print(f"Model: {self.model}")

car1 = Vehicle("Toyota", "Camry")
car2 = Vehicle("Honda", "Civic")

print(Vehicle.vehicle_type)

car1.display_info()
car2.display_info()
```

```
Car
Vehicle Type: Car
Brand: Toyota
Model: Camry
Vehicle Type: Car
Brand: Honda
Model: Civic
```