Assignment 1

July 29, 2023

1 Q1. Create one variable containing following type of data:

(i) string

(ii) list

(iii) float

(iv) tuple

Ans: apple <class 'str'> [2, 3, 4] <class 'list'> 2.125 <class 'float'> ('a', 'b') <class 'tuple'>

[24]:
```
a="apple",[2,3,4],2.125,('a','b')
for i in a:
    print(i,type(i))
```

```
apple <class 'str'>
[2, 3, 4] <class 'list'>
2.125 <class 'float'>
('a', 'b') <class 'tuple'>
```

2 Q2. Given are some following variables containing data:

(i) var1 = ' '

(ii) var2 = '[DS , ML , Python]'

(iii) var3 = ['DS' , 'ML' , 'Python']

(iv) var4 = 1.

Ans: ' '<class 'str'> [DS,ML,Python] <class 'str'> ['DS', 'ML', 'Python'] <class 'list'> 1.0 <class 'float'>

[25]:
```
var1=''
print(var1,type(var1))
var2='[DS,ML,Python]'
print(var2,type(var2))
var3=['DS','ML','Python']
print(var3,type(var3))
var4=1.
print(var4,type(var4))
```

```
 <class 'str'>
[DS,ML,Python] <class 'str'>
['DS', 'ML', 'Python'] <class 'list'>
1.0 <class 'float'>
```

3 Q3. Explain the use of the following operators using an example:

(i) /

(ii) %

(iii) //

(iv) **

Ans: / : this is the division operator

```
%  : this is the Modulus operator

// : this is the Floor division operator

** : this is the Exponentiation operator
```

[26]:
```
a=15
b=5
print (a/b)
print("this is the division operator")
```

```
3.0
this is the division operator
```

[27]:
```
a=15
b=5
print (a%b)
print("this is the Modulus operator")
```

```
0
this is the Modulus operator
```

[28]:
```
a=15
b=5
print (a//b)
print("this is the Floor division operator")
```

```
3
this is the Floor division operator
```

[29]:
```
a=2
b=3
print (a**b)
```

```
print("this is the Exponentiation operator")
```

```
8
this is the Exponentiation operator
```

4 Q4. Create a list of length 10 of your choice containing multiple types of data. Using for loop print the element and its data type.

Ans:2 <class 'int'> 3 <class 'int'> 4 <class 'int'> 5 <class 'int'> 2.325 <class 'float'> True <class 'bool'> ['a', 'b', 'c'] <class 'list'> money <class 'str'> ('w', 't', 'u') <class 'tuple'> False <class 'bool'>

[30]:
```
l=[2,3,4,5,2.325,True,['a','b','c'],"money",("w","t","u"),False]
for a in l:
    print(a,type(a))
```

```
2 <class 'int'>
3 <class 'int'>
4 <class 'int'>
5 <class 'int'>
2.325 <class 'float'>
True <class 'bool'>
['a', 'b', 'c'] <class 'list'>
money <class 'str'>
('w', 't', 'u') <class 'tuple'>
False <class 'bool'>
```

5 Q5. Using a while loop, verify if the number A is purely divisible by number B and if so then how many times it can be divisible.

Ans: Enter your First number, 100. Enter your Second number, 10. The First number is divisible by the Second number, The First number is divisile 2 times, The updated value of the number is 1.

[33]:
```
print("Enter your First number")
a=int(input())

print("Enter your Second number")
b=int(input())

if b==0:
    print("it cannot be divided by zero, please enter a new number ")
else:
    count=0
    while a%b==0:
        count += 1
```

```
        a=a//b
    if count>0:
        print("The First number is divisible by the Second number")
        print("The First number is divisile",count,"times")
        print("The updated value of the number is",a)
    else:
        print("The First number is not divisible by Second number")
```

```
Enter your First number

 777

Enter your Second number

 7

The First number is divisible by the Second number
The First number is divisile 1 times
The updated value of the number is 111
```

6 Q6. Create a list containing 25 int type data. Using for loop and if-else condition print if the element is divisible by 3 or not.

Ans: It Has 25 int type data and it shows which are divisible or not divisible by 3.

[5]:
```
␣
  ↪l=[12,24,55,4,54,54,4884,851,84,1,841,6485,54,51,99,212,2354,221,2548,4785,6658,58,557,995,
for i in l:
    if i%3==0:
        print(i,"is divisible by 3")
    else:
        print(i,"is not divisible by 3")
```

```
12 is divisible by 3
24 is divisible by 3
55 is not divisible by 3
4 is not divisible by 3
54 is divisible by 3
54 is divisible by 3
4884 is divisible by 3
851 is not divisible by 3
84 is divisible by 3
1 is not divisible by 3
841 is not divisible by 3
6485 is not divisible by 3
54 is divisible by 3
51 is divisible by 3
99 is divisible by 3
212 is not divisible by 3
```

```
2354 is not divisible by 3
221 is not divisible by 3
2548 is not divisible by 3
4785 is divisible by 3
6658 is not divisible by 3
58 is not divisible by 3
557 is not divisible by 3
995 is not divisible by 3
782 is not divisible by 3
```

7 Q7. What do you understand about mutable and immutable data types? Give examples for both showing this property.

Ans: Mutable data types are those that can be modified after they are created. This means that the value of a mutable obejct can be changed without creating a new object.

Immutable data types are those that cannot be modified after they are created. This means that if the value of an immutable object, we need to create a new object with the desired value example of immutable data types are integer, floats, tuples.

[13]:
```
Mutable=[2,3,4,5,6,7,8,9]
print(Mutable)
```

```
[2, 3, 4, 5, 6, 7, 8, 9]
```

[18]:
```
Mutable[2]="a"
```

[19]:
```
print(Mutable)
```

```
[2, 3, 'a', 5, 6, 7, 8, 9]
```

[20]:
```
Mutable.append("apple")
```

[21]:
```
print(Mutable)
```

```
[2, 3, 'a', 5, 6, 7, 8, 9, 'apple']
```

[22]:
```
Immutable="money"
print(Immutable)
```

```
money
```

[23]:
```
nstr= Immutable + " shot"
print(nstr)
```

```
money shot
```

[ ]:
```

```