ASSIGNMENT 1

July 14, 2023

[ ]:
```
Q1. Create one variable containing following type of data:
(i) string
(ii) list
(iii) float
(iv) tuple
```

[1]:
```
name = "Muskan"
type(name)
```

[1]:
```
str
```

[2]:
```
mlist = ['pencil','pen','eraser']
type(mlist)
```

[2]:
```
list
```

[4]:
```
num = 6.89
type(num)
```

[4]:
```
float
```

[7]:
```
names = ('anjana','priya')
type(names)
```

[7]:
```
tuple
```

[ ]:
```
Q2. Given are some following variables containing data:
(i) var1 = ‘ ‘
(ii) var2 = ‘[ DS , ML , Python]’
(iii) var3 = [ ‘DS’ , ’ML’ , ‘Python’ ]
(iv) var4 = 1.

What will be the data type of the above given variable.
```

[8]:
```
var1 =''
type(var1)
```

[8]:
```
str
```

[9]:
```
var2 = '[DS,ML,Python]'
type(var2)
```

[9]: str

[10]:
```
var3 = ['DS','ML','Python']
type(var3)
```

[10]: list

[11]:
```
var4 = 1
type(var4)
```

[11]: int

[ ]:
```
Q3. Explain the use of the following operators using an example:
(i) /
(ii) %
(iii) //
(iv) **
```

[ ]:
```
/ : It's a Division operator which gives the quotient
```

[13]:
```
6/2
```

[13]: 3.0

[ ]:
```
 % : It's a modulus operator which gives the remainder
```

[14]:
```
4%5
```

[14]: 4

[ ]:
```
// : It's a floor division operator which is used to find the floor of the␣
 ↪quotient
```

[15]:
```
10//2
```

[15]: 5

[ ]:
```
** : It's a exponentiation operator. It is used to raise the first operand to␣
 ↪the power of the second.
```

[16]:
```
2**3
```

[16]: 8

[ ]:
```
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.
```

[23]:
```
mixed_list=['orange',1,1.036,True,['a','b'],{'name':'priya','gender':␣
 ↪'Female'},('hotel','school','park'),'Neha',0,2+6j]
for i in mixed_list:
  print(type(i))
```

```
<class 'str'>
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'list'>
<class 'dict'>
<class 'tuple'>
<class 'str'>
<class 'int'>
<class 'complex'>
```

[ ]:
```
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.
```

[31]:
```
A=6
B=2
while A%B==0:
  print('A is completely divisible by B')
  print(f'A is divisble by B by {A/B} no of times')
  break
else:
  print('A is not completely divisible by B')
```

```
A is completely divisible by B
A is divisble by B by 3.0 no of times
```

[ ]:
```
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.
```

[32]:
```
plist=[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]
for i in plist:
  if i%3==0:
   print(f'{i} is divisible by 3')
  else:
     print(f'{i} is not divisible by 3')
```

```
1 is not divisible by 3
2 is not divisible by 3
3 is divisible by 3
4 is not divisible by 3
5 is not divisible by 3
6 is divisible by 3
7 is not divisible by 3
8 is not divisible by 3
9 is divisible by 3
10 is not divisible by 3
11 is not divisible by 3
12 is divisible by 3
13 is not divisible by 3
14 is not divisible by 3
15 is divisible by 3
16 is not divisible by 3
17 is not divisible by 3
18 is divisible by 3
19 is not divisible by 3
20 is not divisible by 3
21 is divisible by 3
22 is not divisible by 3
23 is not divisible by 3
24 is divisible by 3
25 is not divisible by 3
```

[ ]:
```
Q7. What do you understand about mutable and immutable data types? Give␣
  ↪examples for both showing
this property.
```

[35]:
```
l1 = [1,2,3,4,5]
print(l1)
l1.insert(5, 6)
print(l1)
```

```
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 6]
```

[36]:
```
tuple = (1,2,3,4,5)
print(tuple)
```

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

[ ]:
```
Mutable--> An python object that has capacity to modify its value is know as␣
  ↪mutable. For exampleList, Dictionaries, Sets.
```

[ ]:
```
Immutable--> An python object that cannot modify its value is know as imutable.␣
 ↪The value of these python object is
fixed once they are made. For example Tuples,String.
```

[ ]:
```

```

[ ]:
```

```