University Institute of Engineering

Department of Computer Science & Engineering

DISRUPTIVE TECHNOLOGY-1
WORKSHEET

Project: Different data types using python

Student Name: Ayush Saxena

UID: 22BCS10778

Branch: Computer Science & Engineering

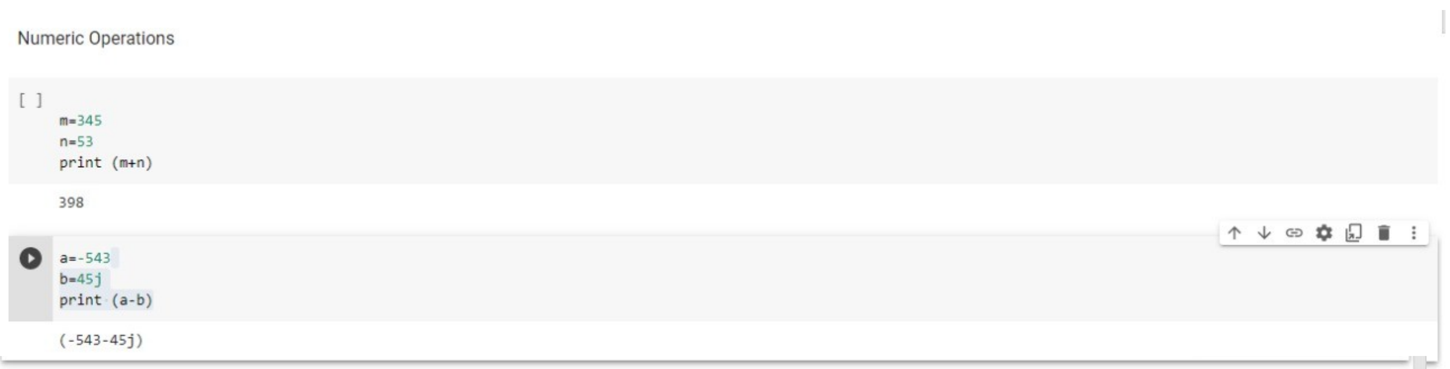
Section/Group: 22BCS212-A

Semester: First

Date of Performance: 02/09/22

Subject Name: DISRUPTIVE TECHNOLOGY-1

Subject Code: 22ECH-102



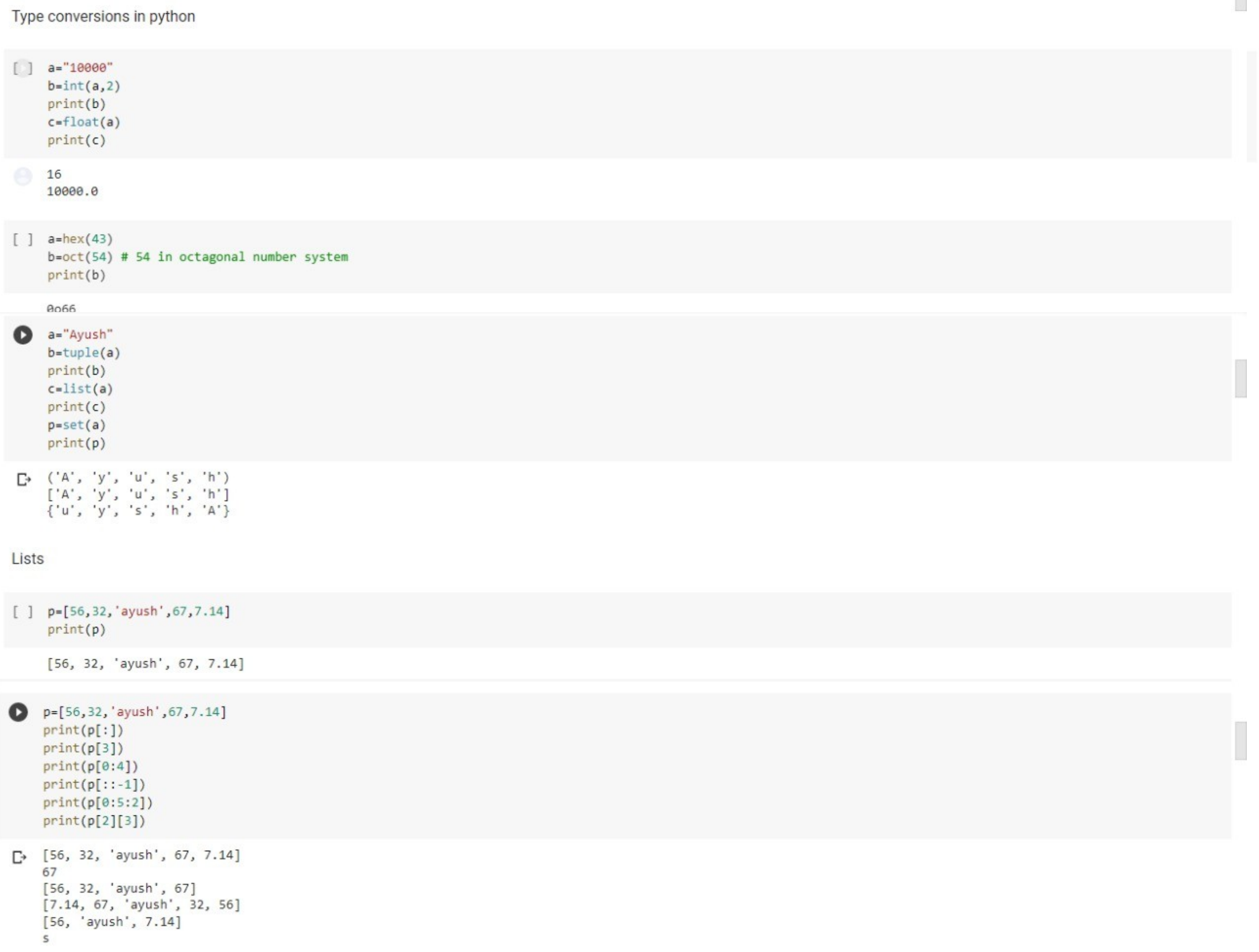
Numeric Operations

```
[ ]
m=345
n=53
print (m+n)
```

```
398
```

```
a=-543
b=45j
print (a-b)
```

```
(-543-45j)
```



Type conversions in python

```
[ ] a="10000"
    b=int(a,2)
    print(b)
    c=float(a)
    print(c)
```

```
16
10000.0
```

```
[ ] a=hex(43)
    b=oct(54) # 54 in octagonal number system
    print(b)
```

```
0o66
```

```
a="Ayush"
b=tuple(a)
print(b)
c=list(a)
print(c)
p=set(a)
print(p)
```

```
('A', 'y', 'u', 's', 'h')
['A', 'y', 'u', 's', 'h']
{'u', 'y', 's', 'h', 'A'}
```

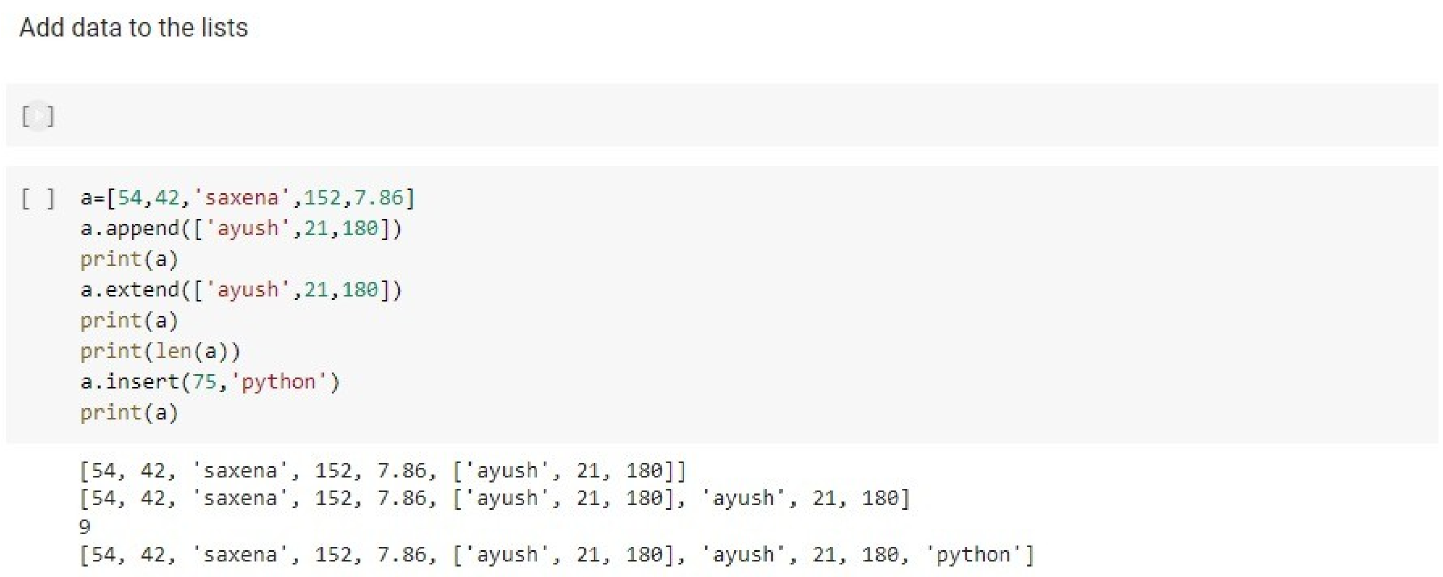
Lists

```
[ ] p=[56,32,'ayush',67,7.14]
    print(p)
```

```
[56, 32, 'ayush', 67, 7.14]
```

```
p=[56,32,'ayush',67,7.14]
print(p[:])
print(p[3])
print(p[0:4])
print(p[::-1])
print(p[0:5:2])
print(p[2][3])
```

```
[56, 32, 'ayush', 67, 7.14]
67
[56, 32, 'ayush', 67]
[7.14, 67, 'ayush', 32, 56]
[56, 'ayush', 7.14]
s
```

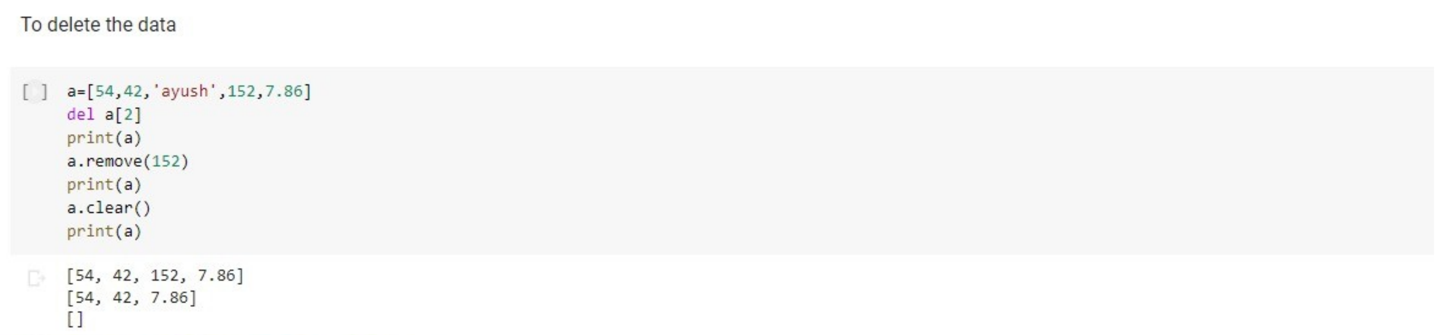


Add data to the lists

```
[ ]
```

```
[ ] a=[54,42,'saxena',152,7.86]
    a.append(['ayush',21,180])
    print(a)
    a.extend(['ayush',21,180])
    print(a)
    print(len(a))
    a.insert(75,'python')
    print(a)
```

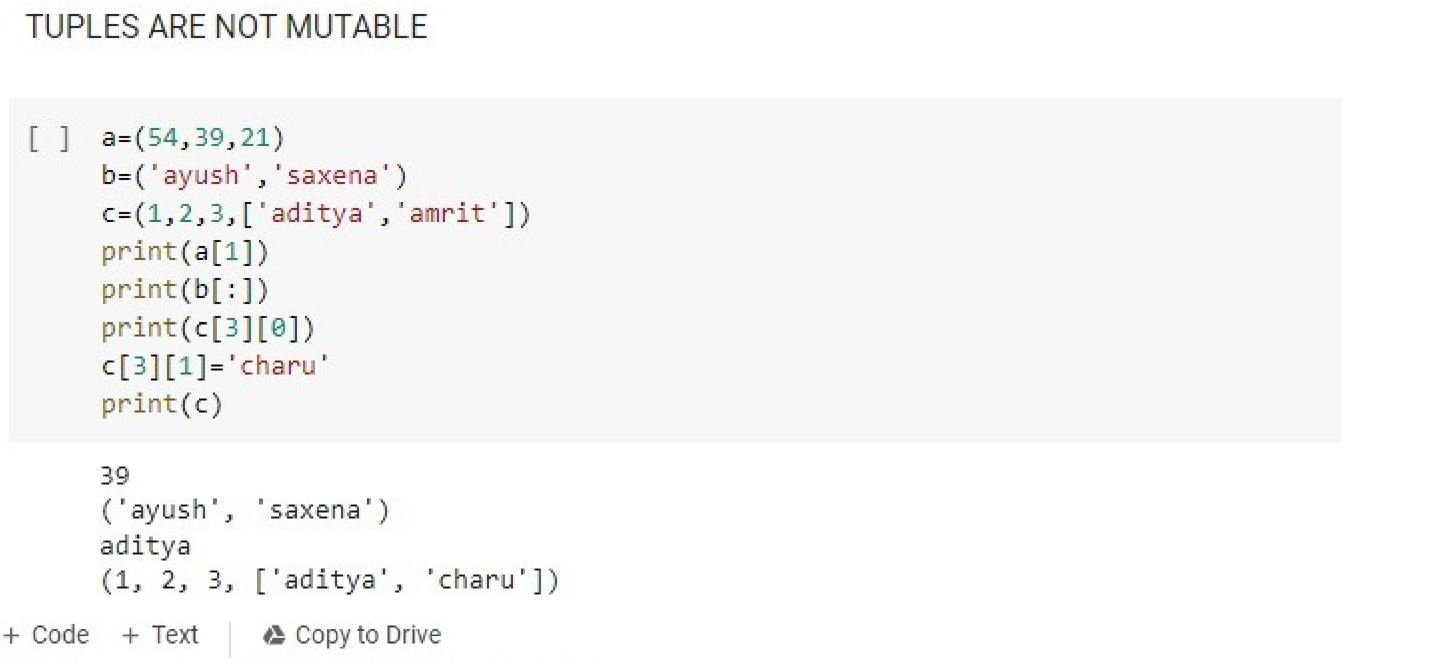
```
[54, 42, 'saxena', 152, 7.86, ['ayush', 21, 180]]
[54, 42, 'saxena', 152, 7.86, ['ayush', 21, 180], 'ayush', 21, 180]
9
[54, 42, 'saxena', 152, 7.86, ['ayush', 21, 180], 'ayush', 21, 180, 'python']
```



To delete the data

```
a=[54,42,'ayush',152,7.86]
del a[2]
print(a)
a.remove(152)
print(a)
a.clear()
print(a)
```

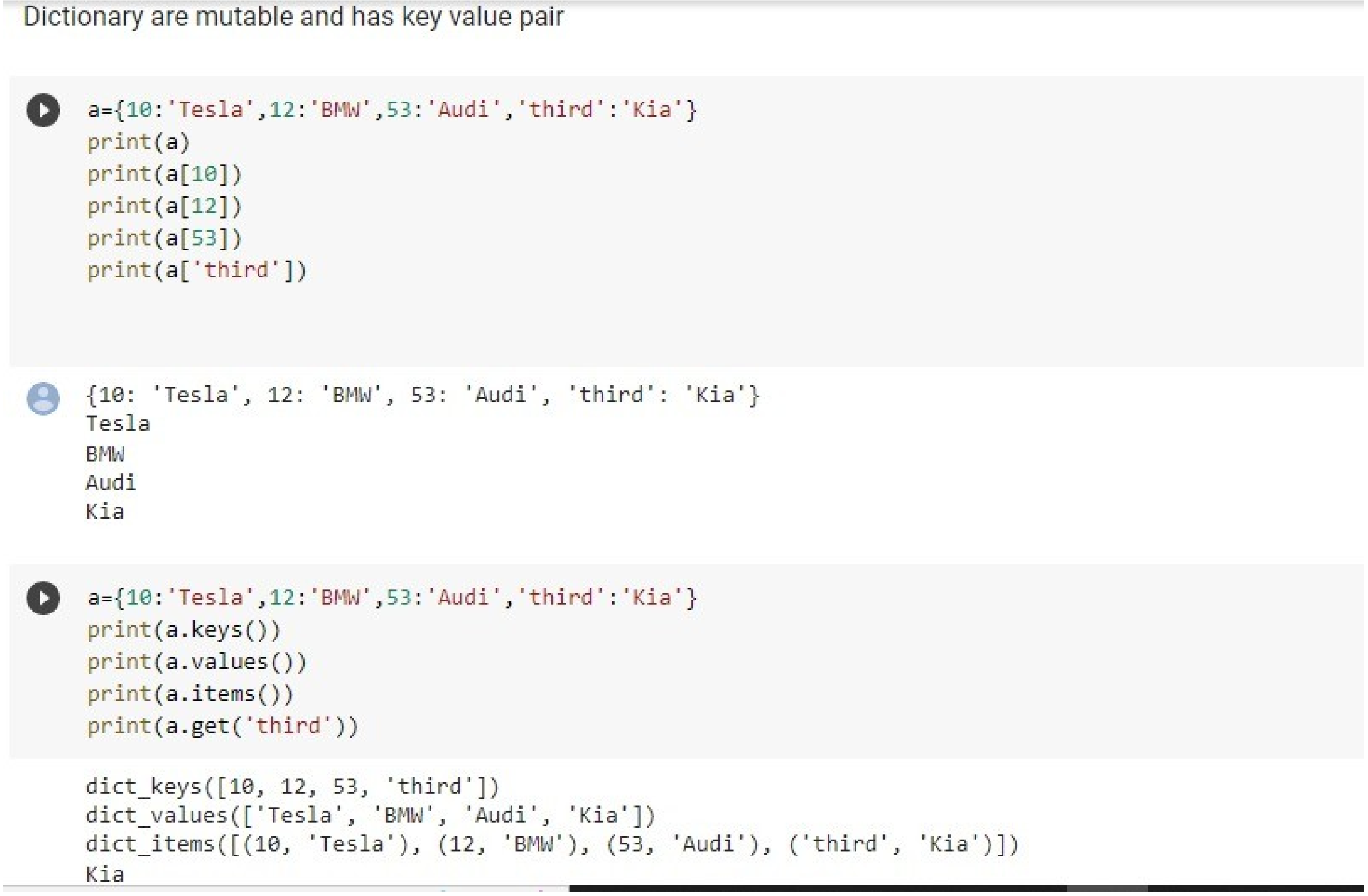
```
[54, 42, 152, 7.86]
[54, 42, 7.86]
[]
```



TUPLES ARE NOT MUTABLE

```
a=(54,39,21)
b=('ayush','saxena')
c=(1,2,3,['aditya','amrit'])
print(a[1])
print(b[:])
print(c[3][0])
c[3][1]='charu'
print(c)
```

```
39
('ayush', 'saxena')
aditya
(1, 2, 3, ['aditya', 'charu'])
```



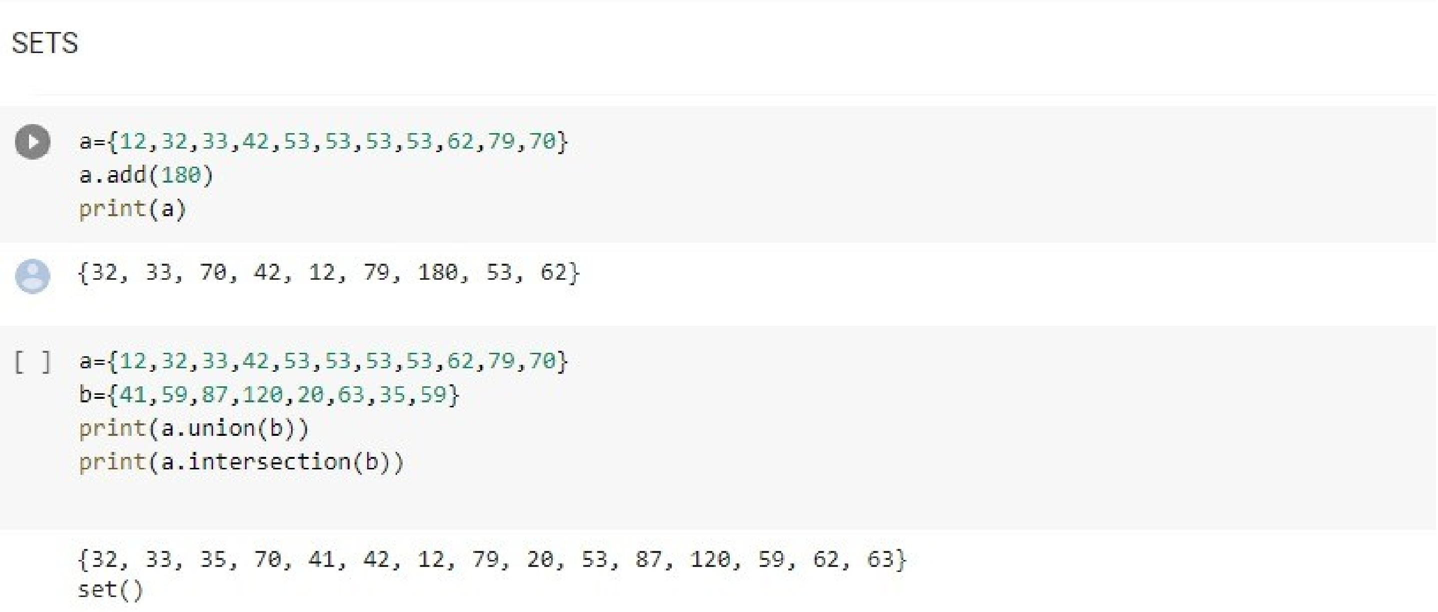
Dictionary are mutable and has key value pair

```
a={10:'Tesla',12:'BMW',53:'Audi','third':'Kia'}
print(a)
print(a[10])
print(a[12])
print(a[53])
print(a['third'])
```

```
{10: 'Tesla', 12: 'BMW', 53: 'Audi', 'third': 'Kia'}
Tesla
BMW
Audi
Kia
```

```
a={10:'Tesla',12:'BMW',53:'Audi','third':'Kia'}
print(a.keys())
print(a.values())
print(a.items())
print(a.get('third'))
```

```
dict_keys([10, 12, 53, 'third'])
dict_values(['Tesla', 'BMW', 'Audi', 'Kia'])
dict_items([(10, 'Tesla'), (12, 'BMW'), (53, 'Audi'), ('third', 'Kia')])
Kia
```



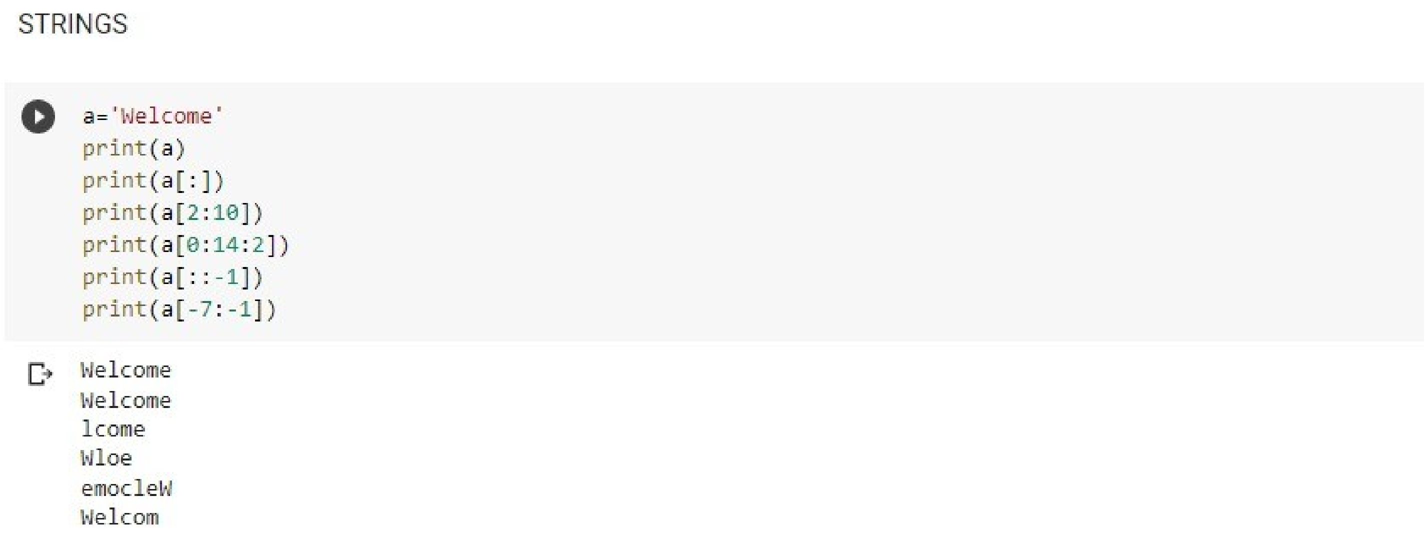
SETS

```
a={12,32,33,42,53,53,53,53,62,79,70}
a.add(180)
print(a)
```

```
{32, 33, 70, 42, 12, 79, 180, 53, 62}
```

```
[ ] a={12,32,33,42,53,53,53,53,62,79,70}
    b={41,59,87,120,20,63,35,59}
    print(a.union(b))
    print(a.intersection(b))
```

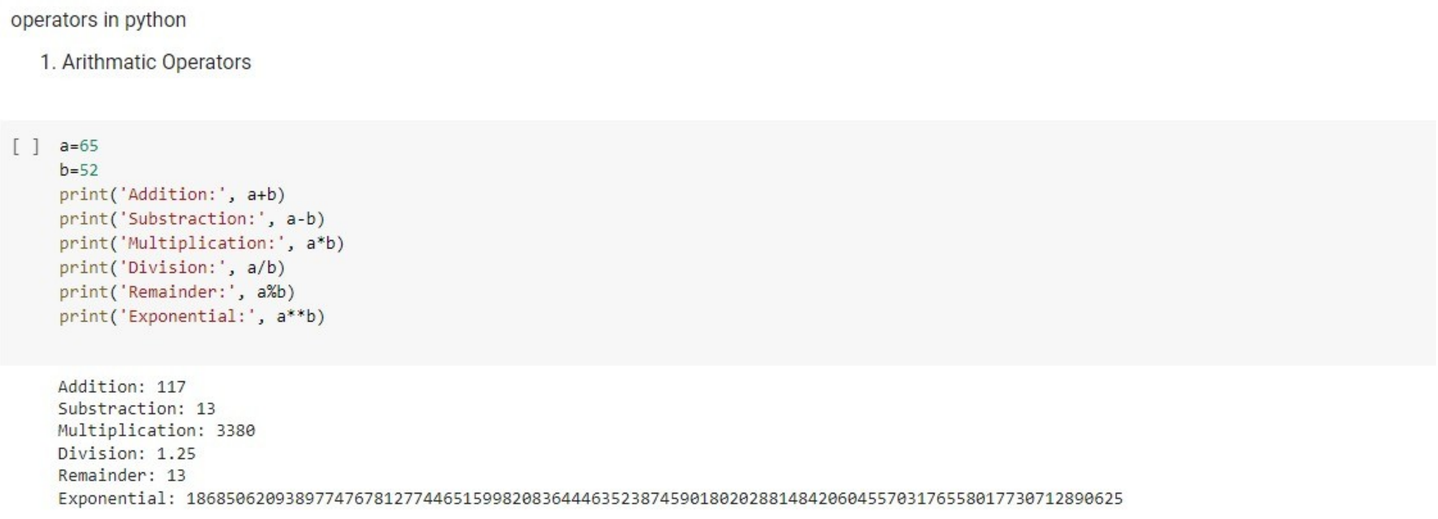
```
{32, 33, 35, 70, 41, 42, 12, 79, 20, 53, 87, 120, 59, 62, 63}
set()
```



STRINGS

```
a='Welcome'
print(a)
print(a[:])
print(a[2:10])
print(a[0:14:2])
print(a[::-1])
print(a[-7:-1])
```

```
Welcome
Welcome
lcome
Wloe
emocleW
Welcom
```

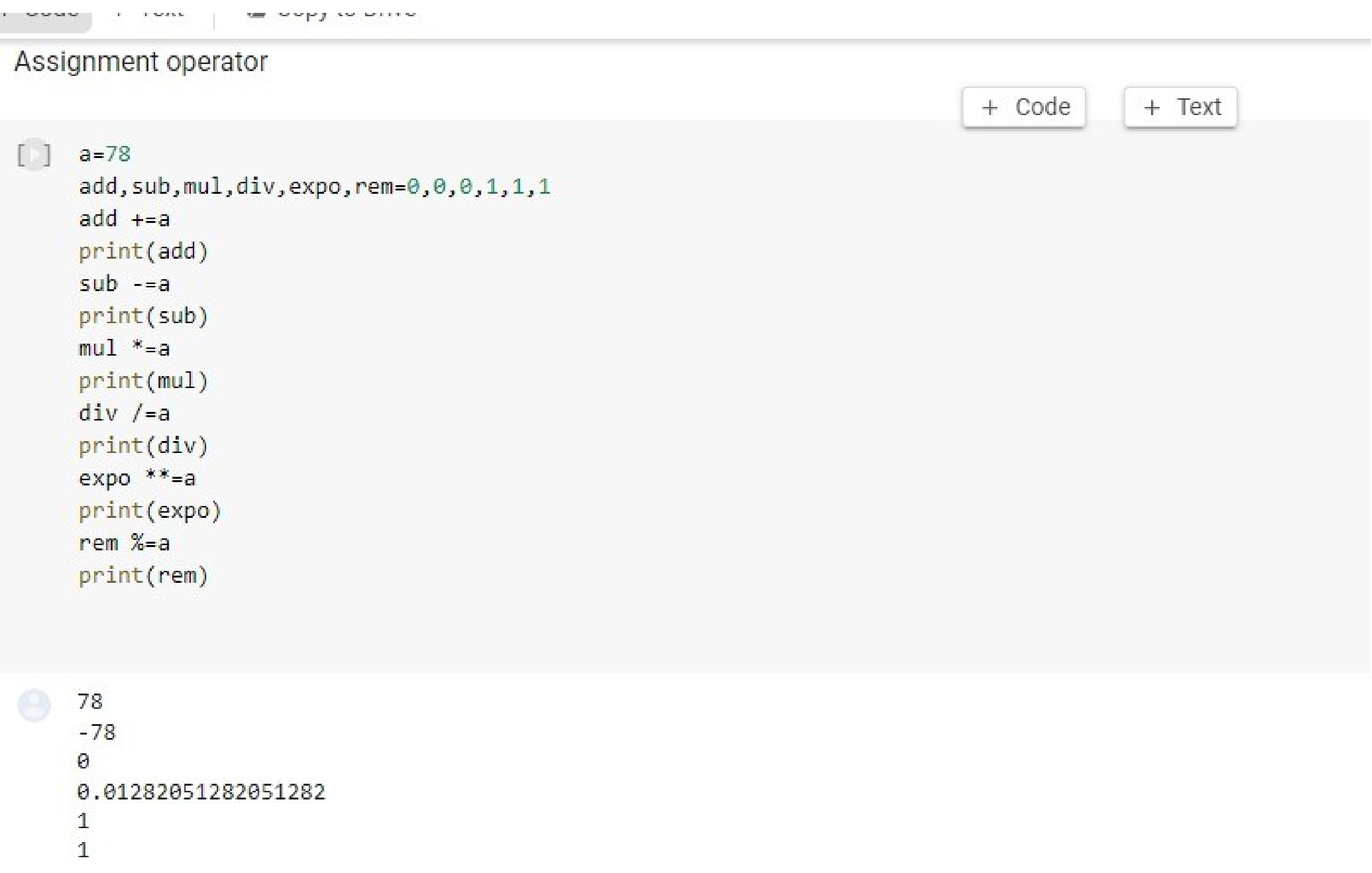


operators in python

1. Arithmatic Operators

```
[ ] a=65
    b=52
    print('Addition:', a+b)
    print('Substraction:', a-b)
    print('Multiplication:', a*b)
    print('Division:', a/b)
    print('Remainder:', a%b)
    print('Exponential:', a**b)
```

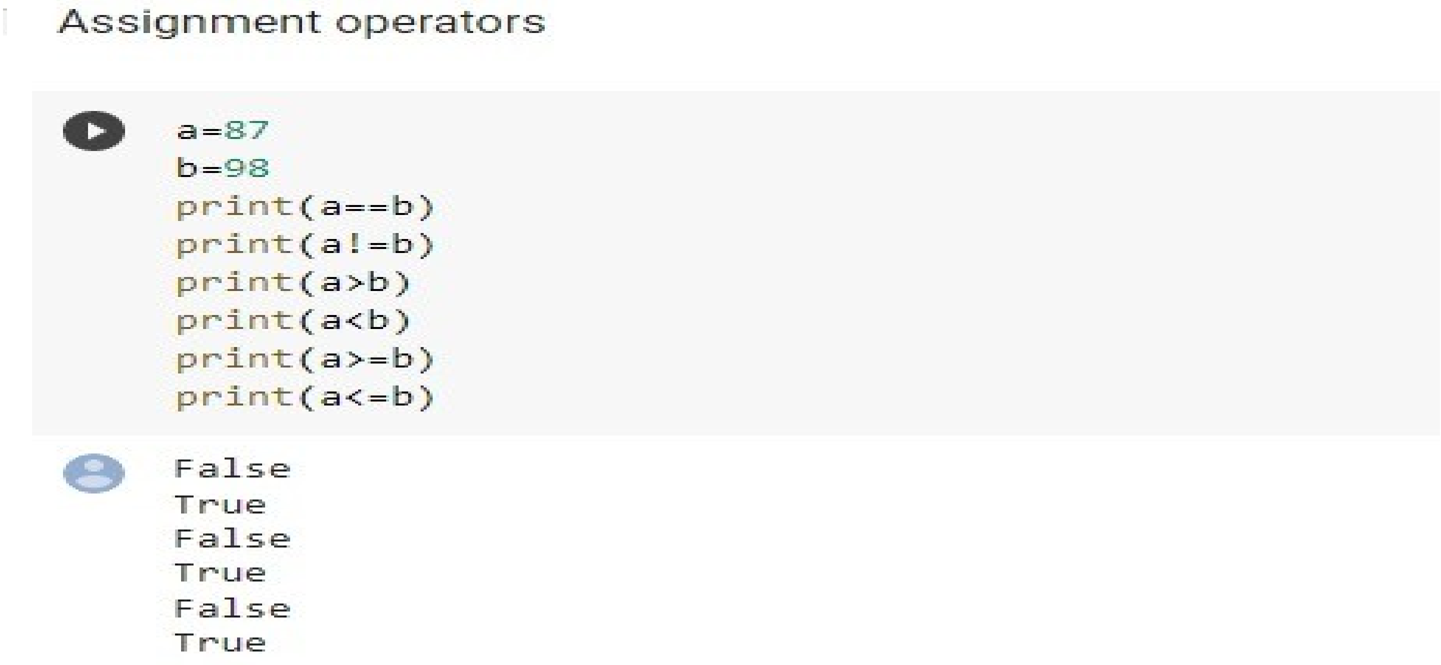
```
Addition: 117
Substraction: 13
Multiplication: 3380
Division: 1.25
Remainder: 13
Exponential: 18685062093897747678127744651599820836444635238745901802028814842060455703176558017730712890625
```



Assignment operator

```
a=78
add,sub,mul,div,expo,rem=0,0,0,1,1,1
add +=a
print(add)
sub -=a
print(sub)
mul *=a
print(mul)
div /=a
print(div)
expo **=a
print(expo)
rem %=a
print(rem)
```

```
78
-78
0
0.01282051282051282
1
1
```

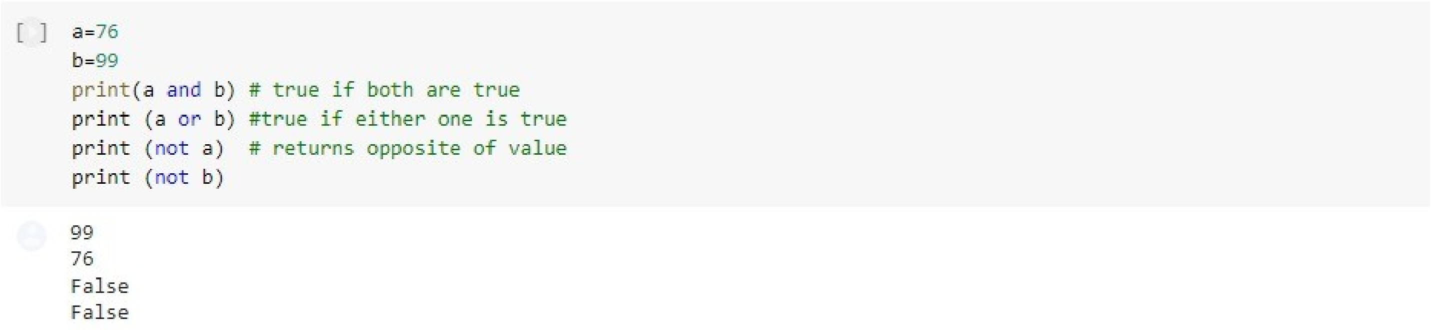


Assignment operators

```
a=87
b=98
print(a==b)
print(a!=b)
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
```

```
False
True
False
True
False
True
```

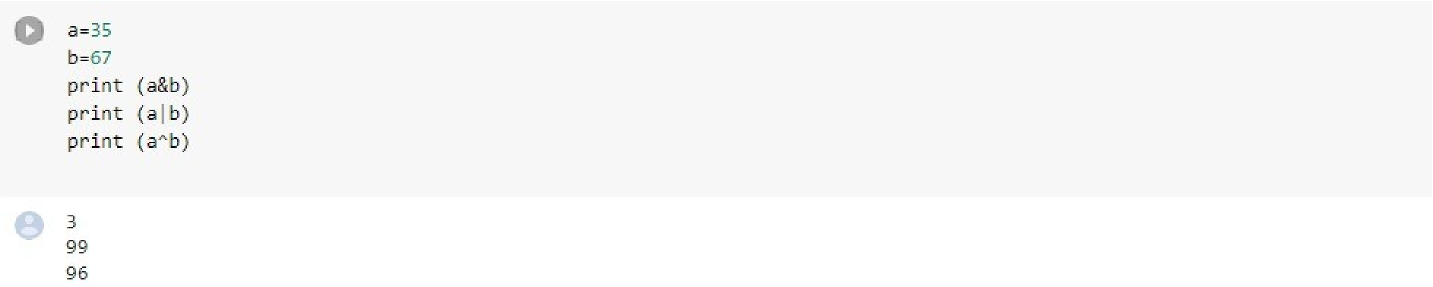
Logical operators



```
a=76
b=99
print(a and b) # true if both are true
print (a or b) #true if either one is true
print (not a)  # returns opposite of value
print (not b)
```

```
99
76
False
False
```

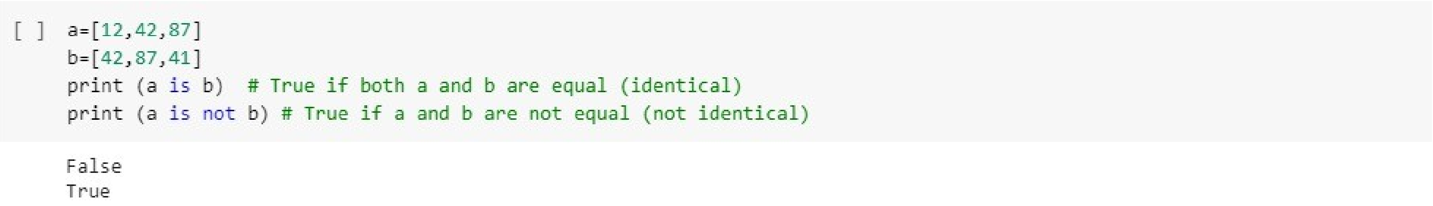
bitwise operator



```
a=35
b=67
print (a&b)
print (a|b)
print (a^b)
```

```
3
99
96
```

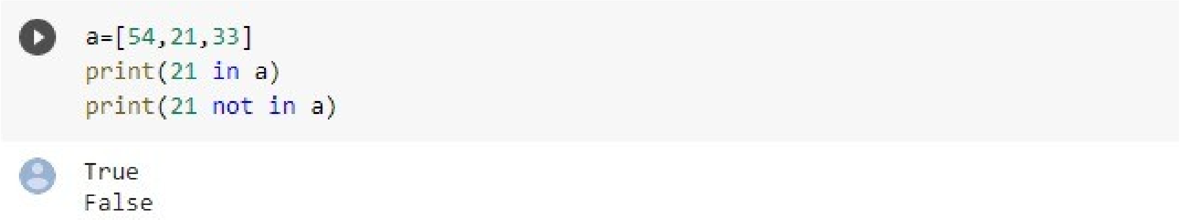
identity operators



```
a=[12,42,87]
b=[42,87,41]
print (a is b)  # True if both a and b are equal (identical)
print (a is not b) # True if a and b are not equal (not identical)
```

```
False
True
```

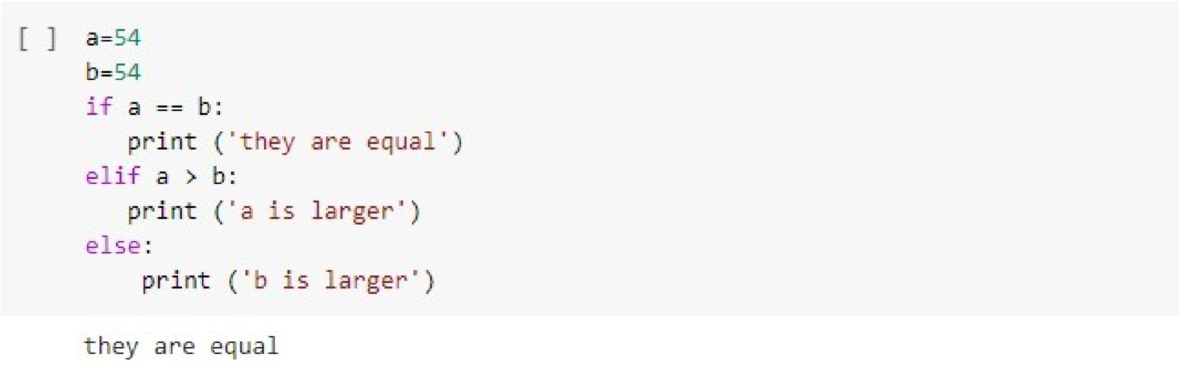
Membership opeartor (value exists or not)



```
a=[54,21,33]
print(21 in a)
print(21 not in a)
```

```
True
False
```

else if condition for flow control



```
a=54
b=54
if a == b:
   print ('they are equal')
elif a > b:
   print ('a is larger')
else:
    print ('b is larger')
```

```
they are equal
```