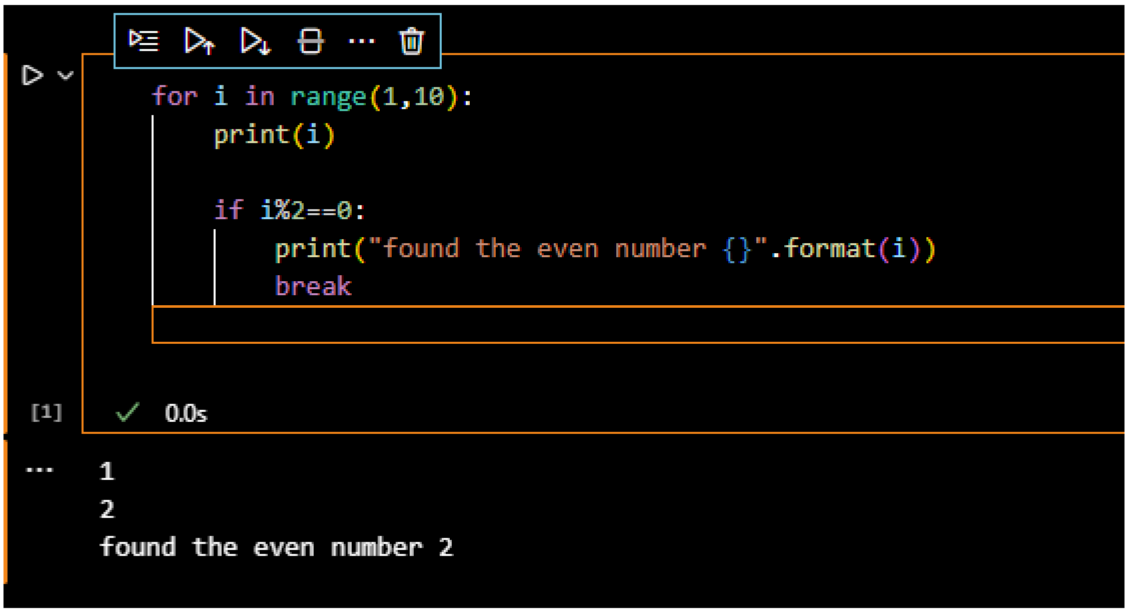
PYTHON ASSIGNMENT 1

1. The main key features of Python that makes it a popular choice for programmers formulate upon the fact that is has a very easy and a simple syntax and it closely resembles the English language. It Is very easy to interpret and supports a wide number of applications ranging from web development, machine learning, deep learning , image & data analysis, data scraping, game development and many more. It has a vast number of libraries and modules which makes python easy to implement these applications. It is a high level and an open source language where one can directly see the code of how it it created. Another one of the main features is that it supports object oriented programming which is a very important methodology now days as it is used for different approaches while working on a project. Thus these factors makes it easy to implement any new technologies and maintain changes in the application for any industry.

2. The predefined keywords in Python play a very important role as these are the words which hold a specific purpose and a meaning for easy interpretability. These words cannot be used as a variable as they are stored differently in relation to the meaning behind them. Python has a lot of words reserved for specific purposes which can be seen below:

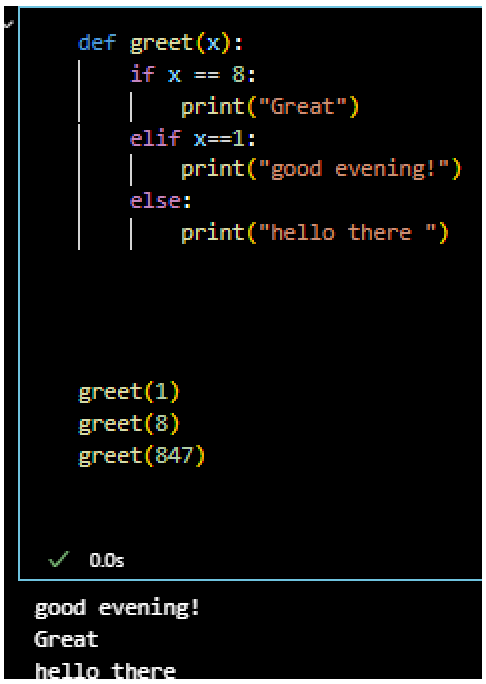


```python
for i in range(1,10):
    print(i)

    if i%2==0:
        print("found the even number {}".format(i))
        break
```

[1] ✓ 0.0s

```
1
2
found the even number 2
```

A. This involves the keywords for,in,break,if. They can be used for loops and checking conditions. We can also use while loop for the same. This code also includes some functions like print,range,format.



```
def greet(x):
    if x == 8:
        print("Great")
    elif x==1:
        print("good evening!")
    else:
        print("hello there ")



greet(1)
greet(8)
greet(847)
```

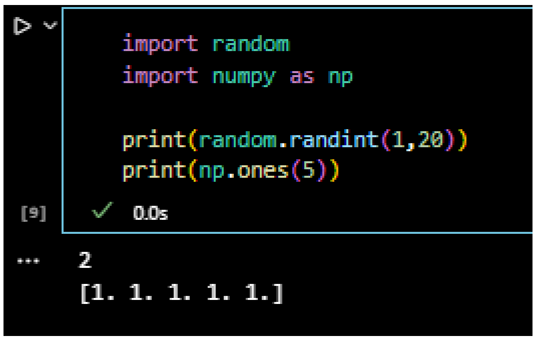
```
✓ 0.0s
good evening!
Great
hello there
```

B. Now this code inlvoves Def which is used for creating a function, if,elif & else, used for check various conditions at once.

C. Similarly “class” is can be used for implementing object oriented progs.

D. Then there are try and except functions which can be used for working with specific errors and handling exceptions.

E. Then comes the import keyword which is used for importing the modules



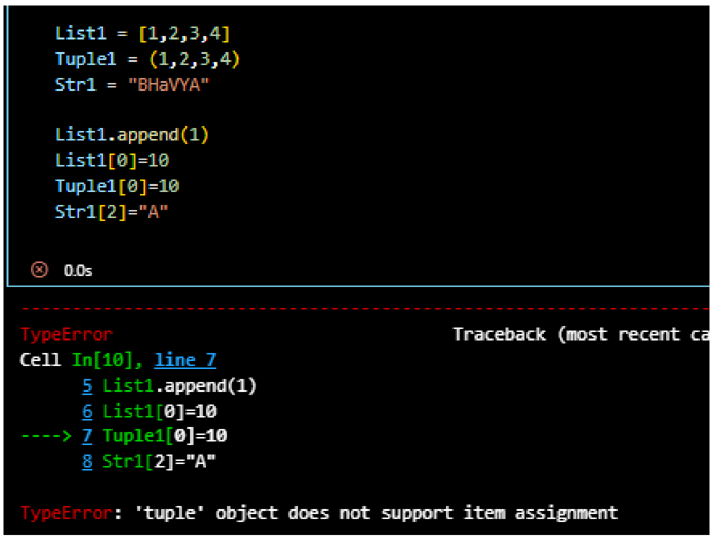
```
import random
import numpy as np

print(random.randint(1,20))
print(np.ones(5))
```

```
[9] ✓ 0.0s
... 2
    [1. 1. 1. 1. 1.]
```

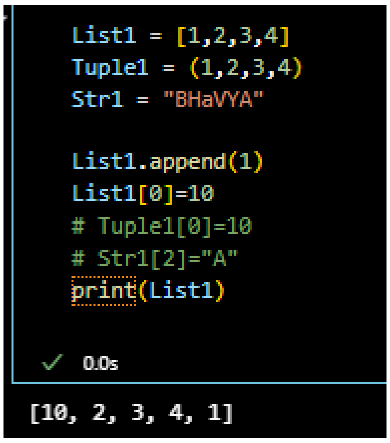
3. Python supports 2 kinds of objects, mutable, meaning which can be modified and immutable, which cant be modified.

   MUTABLE:
   These objects can be changed after their creation and different elements in them can be modified. The whole object is updated and saved. Lists are the most commonly used mutable objects, sets and dictionaries are also mutable.

   IMMUTABLE:
   These objects cannot be modified after their creation. If we want to change a certain element contained inside an immutable oject, there is no way to make changes to it, be it subtracting an element or appending a new one. Tuples and strings lie among the immutable ojects, where if we want to make a change, a whole new string/tuple needs to assigned

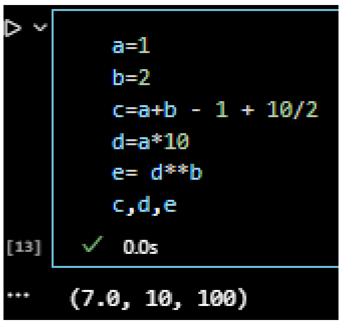


```
List1 = [1,2,3,4]
Tuple1 = (1,2,3,4)
Str1 = "BHaVYA"

List1.append(1)
List1[0]=10
Tuple1[0]=10
Str1[2]="A"

⊗ 0.0s

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent ca
Cell In[10], line 7
      5 List1.append(1)
      6 List1[0]=10
----> 7 Tuple1[0]=10
      8 Str1[2]="A"

TypeError: 'tuple' object does not support item assignment
```

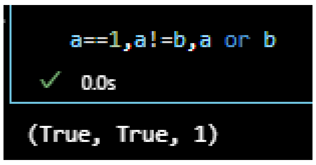


```
List1 = [1,2,3,4]
Tuple1 = (1,2,3,4)
Str1 = "BHaVYA"

List1.append(1)
List1[0]=10
# Tuple1[0]=10
# Str1[2]="A"
print(List1)

✓ 0.0s

[10, 2, 3, 4, 1]
```

4. Python supports a wide range of operators, which can be used for performing arithmetic operations, comparison, asiignments, conditions, etc.

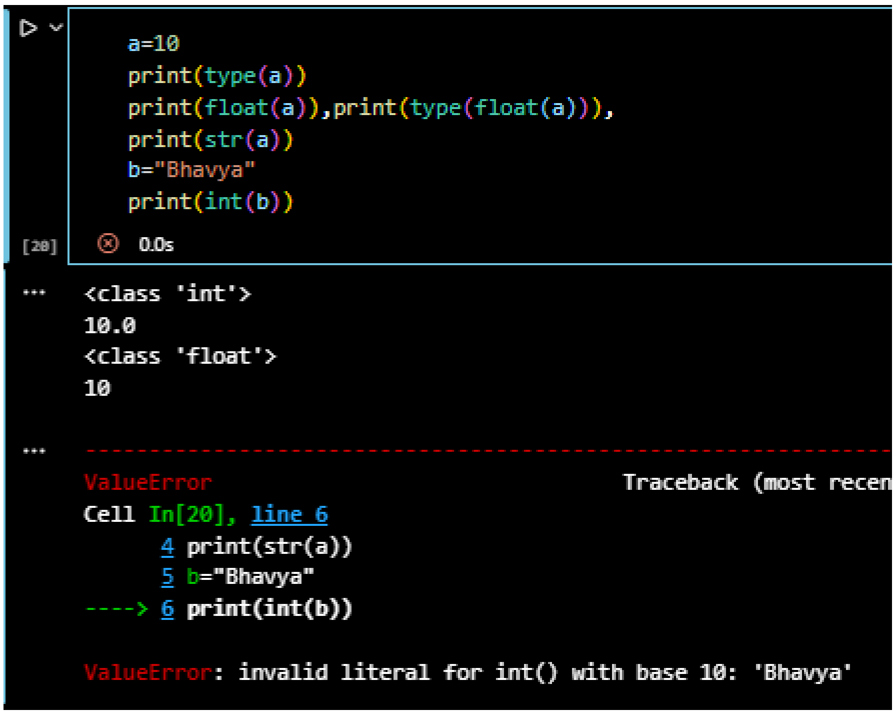


```
a=1
b=2
c=a+b - 1 + 10/2
d=a*10
e= d**b
c,d,e
[13]  ✓ 0.0s

(7.0, 10, 100)
```



```
a==1,a!=b,a or b
✓ 0.0s

(True, True, 1)
```

5. Python langugae supports the conversion of different datatypes. We can convert an integer to a float number or even a string. But you obviosuly cant convert a string of alphabets to an integer. It can be done explicitly and implicitly.

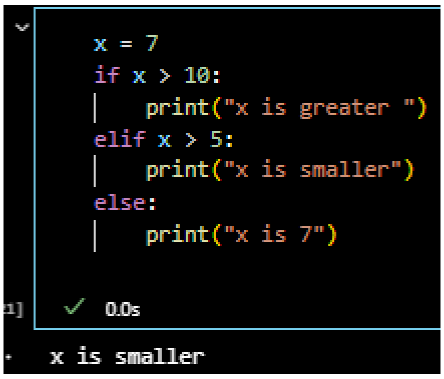


```
a=10
print(type(a))
print(float(a)),print(type(float(a))),
print(str(a))
b="Bhavya"
print(int(b))
```

```
[20]  0.0s
```

```
<class 'int'>
10.0
<class 'float'>
10

---------------------------------------------------------------------------
ValueError                                Traceback (most recen
Cell In[20], line 6
      4 print(str(a))
      5 b="Bhavya"
----> 6 print(int(b))

ValueError: invalid literal for int() with base 10: 'Bhavya'
```

6. Conditional stements work in the same way as they are supposed to, just on the basis of whether a conditon is true or false, the code will undergo the loop or the conditoned mentioned. These are the basic controlling blocks to check any condtion. Based on which ever condition is true unless mentioned, the program moves forward.

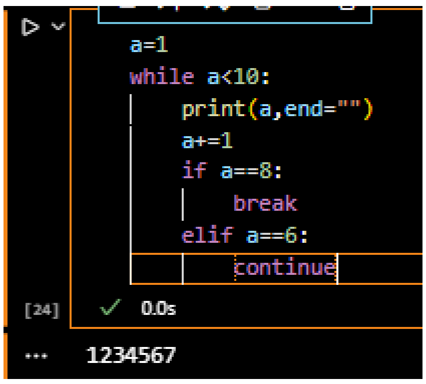


```
x = 7
if x > 10:
    print("x is greater ")
elif x > 5:
    print("x is smaller")
else:
    print("x is 7")
```

```
1]  ✓ 0.0s
```

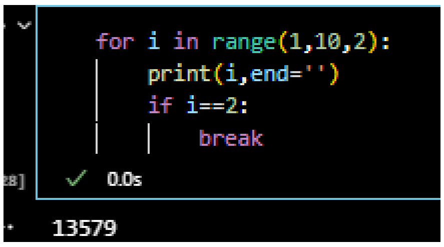
```
x is smaller
```

7. Python has different kinds of loops the simplest of them being a while loop, that runds until the condtion becomes false and if not then it will run infinitely. Then come the for loop, which can be used with the range funtion and will run until the range is met or it can be used to iterate over a list or a string and in such other ways. They can also be compined with a break or continue statement, which as the name suggests will either break the loop when a certain condtion is met or keep iterating.



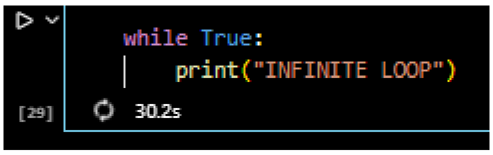
```
a=1
while a<10:
    print(a,end="")
    a+=1
    if a==8:
        break
    elif a==6:
        continue
```

```
[24]  ✓ 0.0s
1234567
```



```
for i in range(1,10,2):
    print(i,end='')
    if i==2:
        break
```

```
8]  ✓ 0.0s
13579
```



```
while True:
    print("INFINITE LOOP")
```

```
[29]  30.2s
```