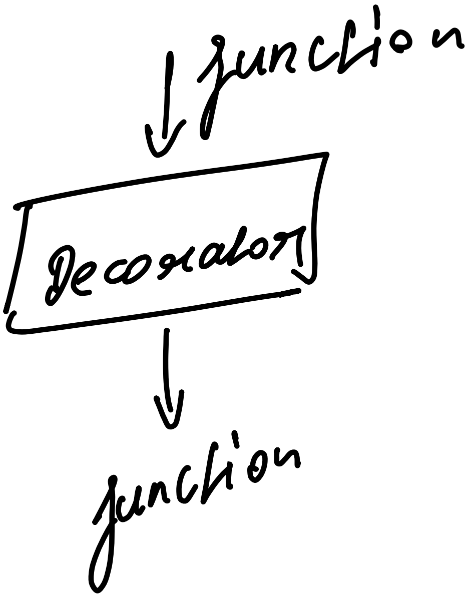
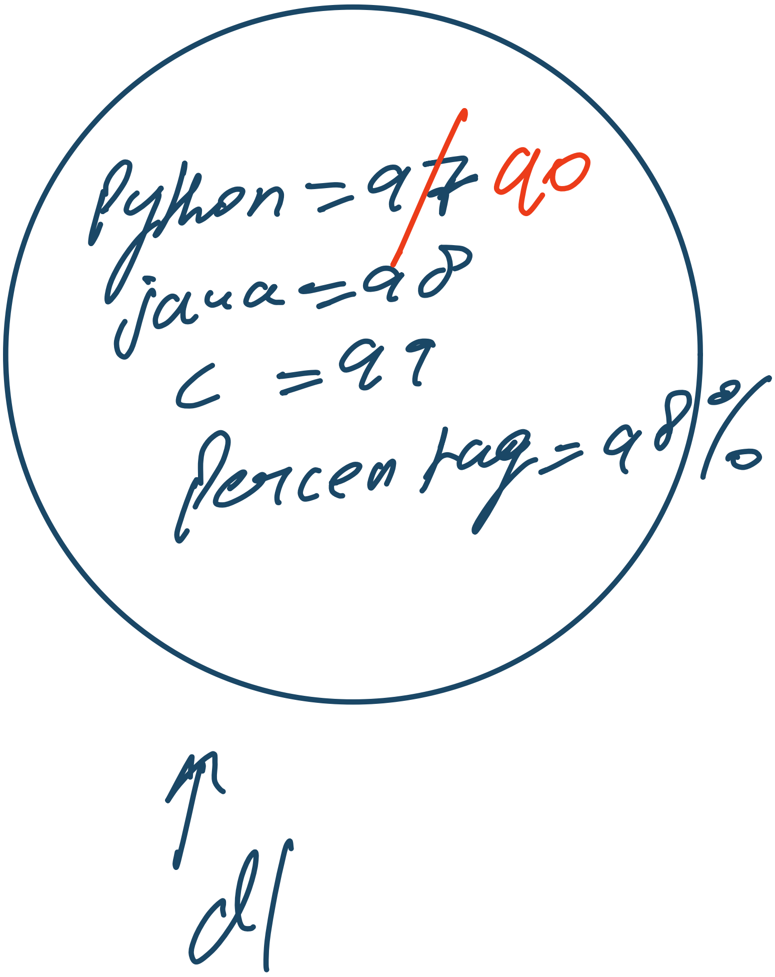
18 Nov

@staticmethod
@classmethod
@property



```
name = "Arun"
age = 28

class Demo :

    def __init__(self, python, java, c):
        ⇒self.python = python
        ⇒ ____
        ⇒ ____
        ⇒self.percentage = str(p+j+c)/3 + "%"   X
    @property
    def    percentage(self) :
             return ⇒


d1=Demo(97,98,99)
    print(d1.percentage)  // 98

d1.python = 90  ⇒
    print(d1.percentage)   // 98
```



python = 9~~7~~ 90

java = 98

c = 97

Percentag = 98%

df

Dunder function

addition ⇒ __add__

subtraction ⇒ __sub__

division ⇒ __truediv__

mul ⇒ __mul__

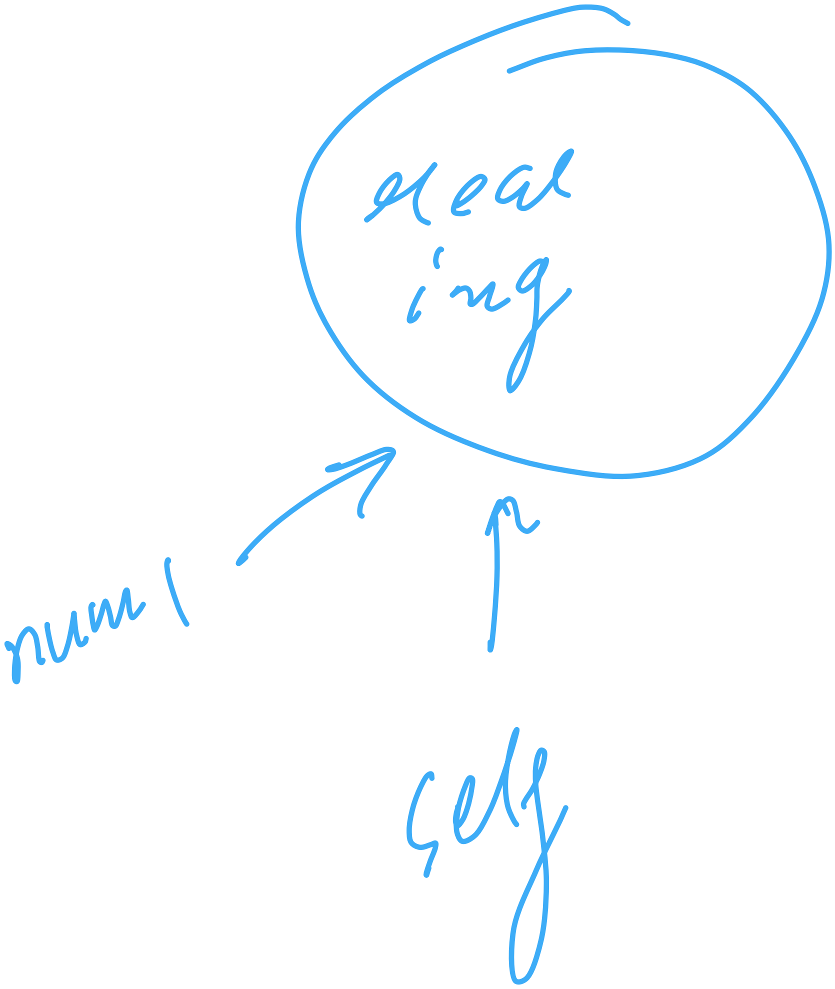
```
class Complex_Number :

        def __init (self, real, img):
            self.real = real
            self.img = img
                                                3i+5j

        def createComplex (self):
            print( real, "i+", img, "j")
                    3i+5j

num1= complex_number(3,5)
c1= num1.createComplex( )
  ⟶       3i+5j
```

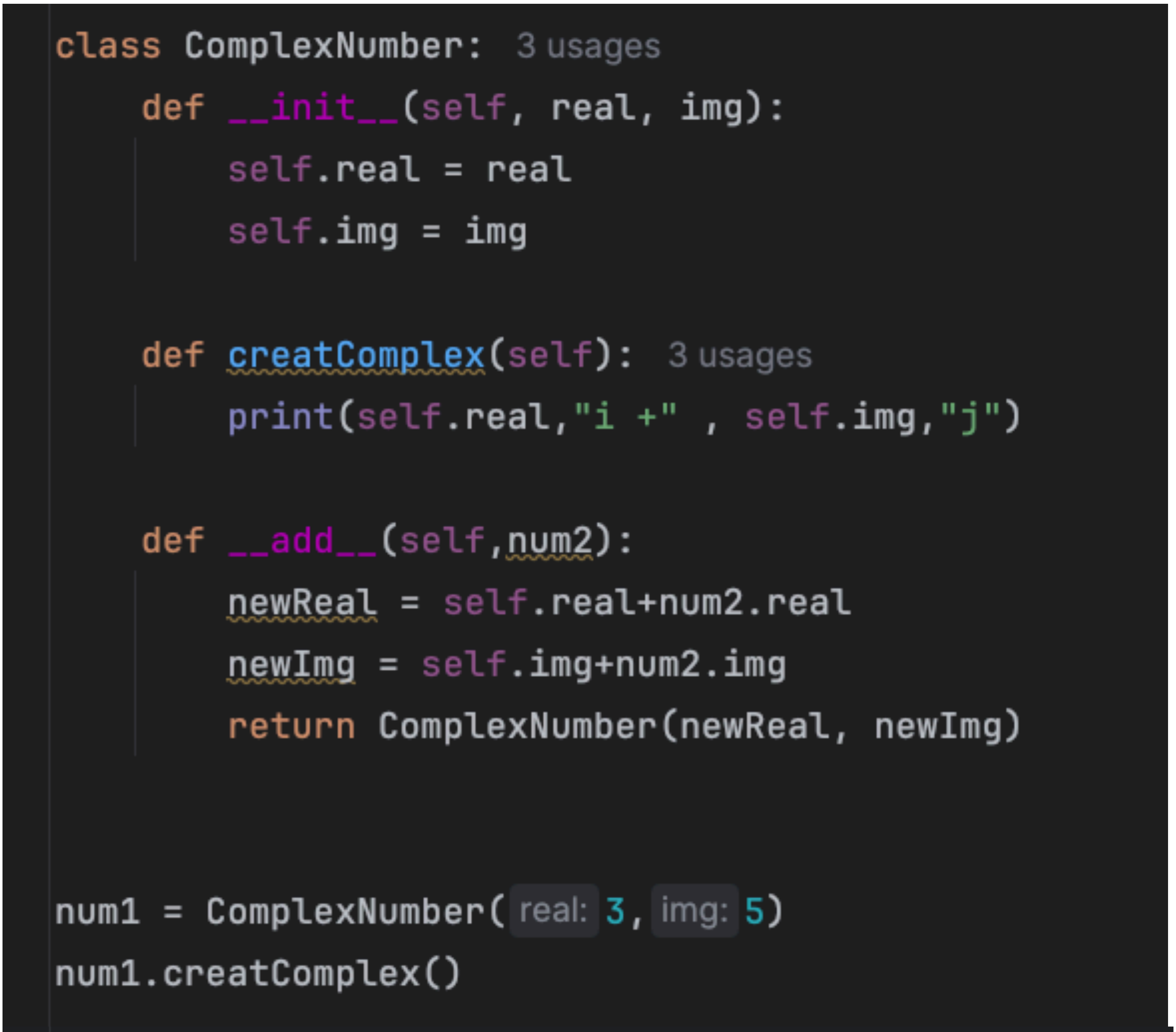
num1 = 3i + 4j

num2 = 2i + 5j

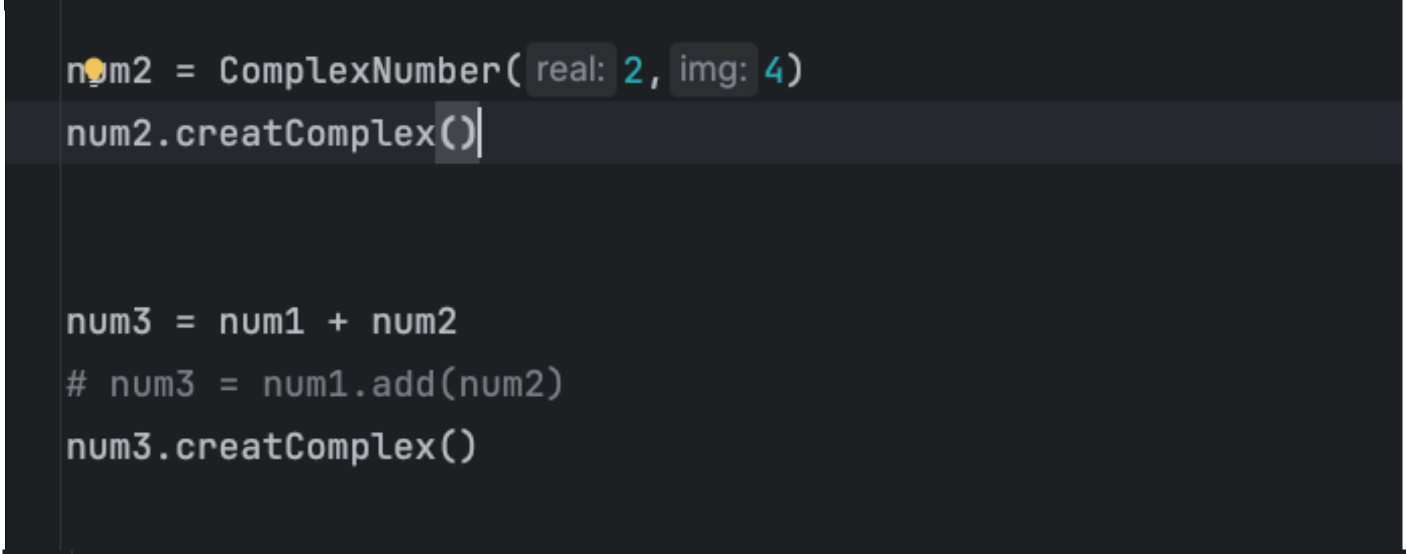
num1 + num2 => 5i + 9j

```
def add(self, num2)
    newReal = self.real + num2.real
    newImg = self.img + num2.img
    return Complex_Number(newReal,
                          newImg)
```

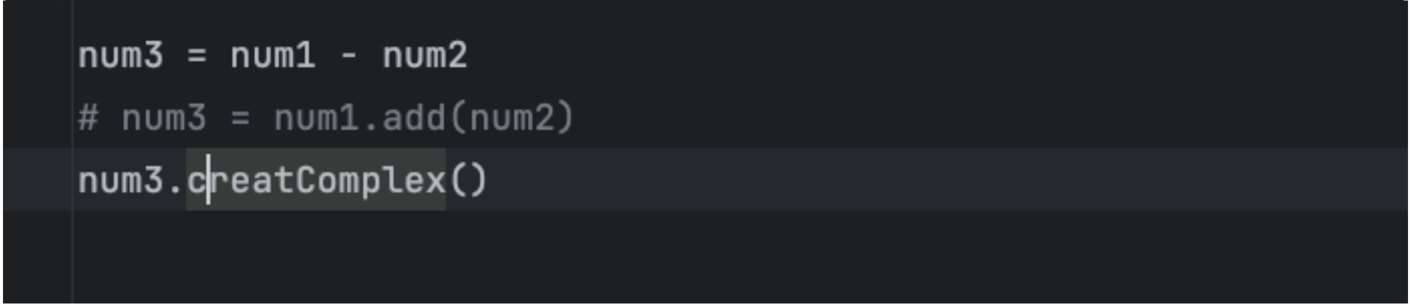

real
img
num1
self



```
class ComplexNumber:  3 usages
    def __init__(self, real, img):
        self.real = real
        self.img = img

    def creatComplex(self):  3 usages
        print(self.real,"i +" , self.img,"j")

    def __add__(self,num2):
        newReal = self.real+num2.real
        newImg = self.img+num2.img
        return ComplexNumber(newReal, newImg)


num1 = ComplexNumber(real: 3, img: 5)
num1.creatComplex()
```



```
num2 = ComplexNumber(real: 2, img: 4)
num2.creatComplex()


num3 = num1 + num2
# num3 = num1.add(num2)
num3.creatComplex()
```

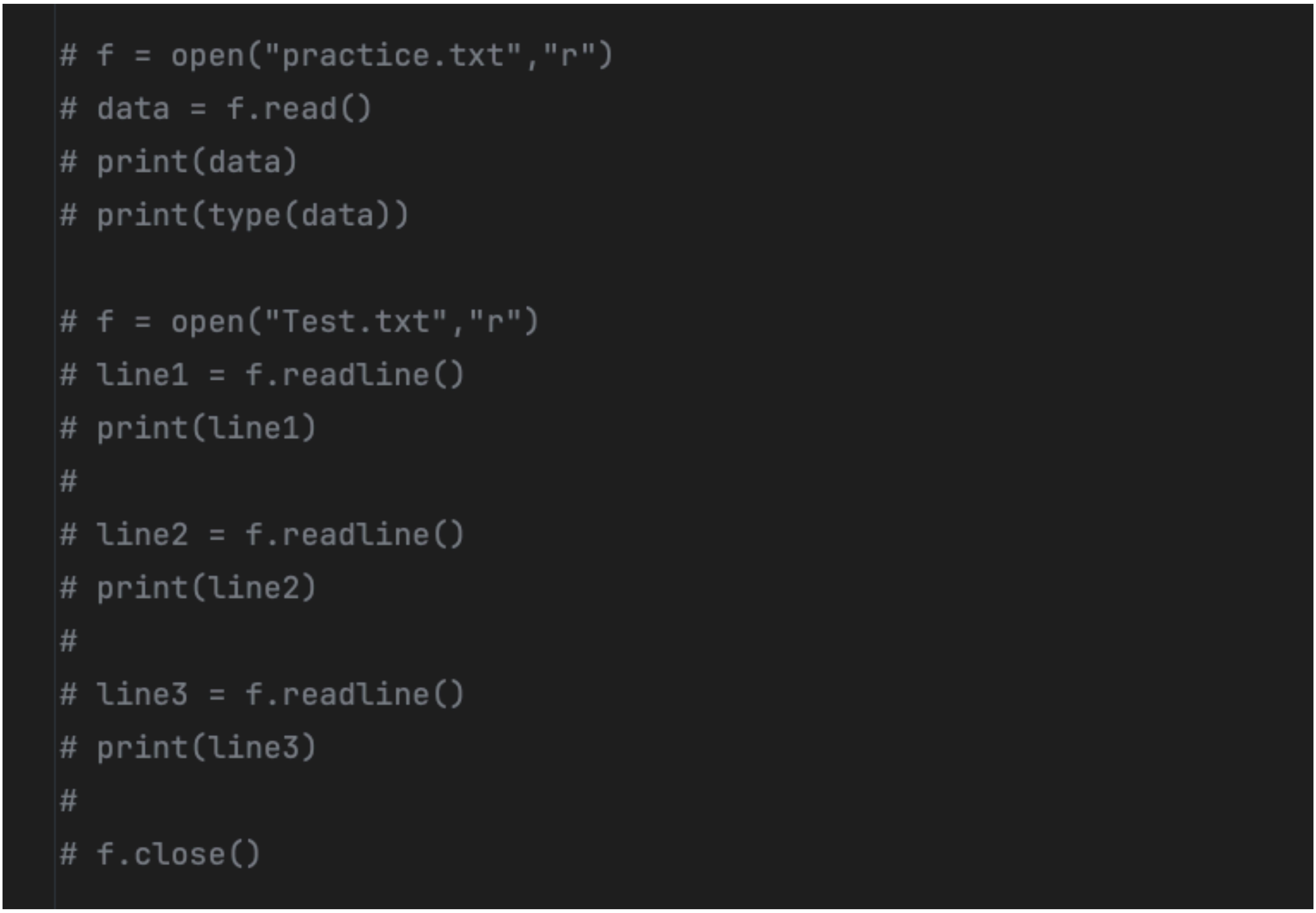


```
num3 = num1 - num2
# num3 = num1.add(num2)
num3.creatComplex()
```

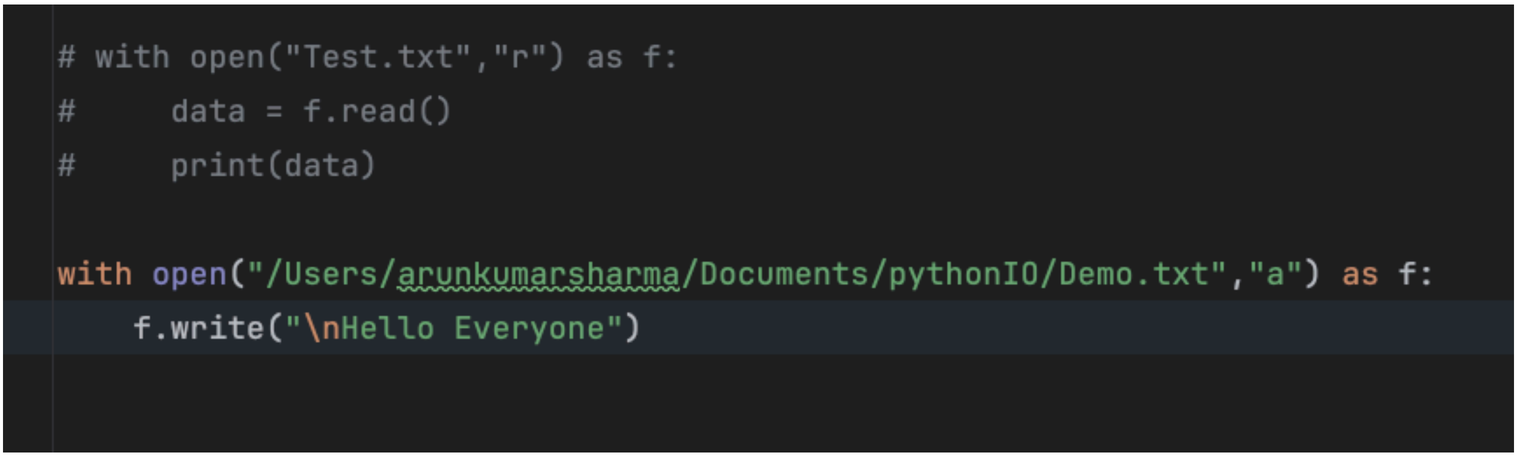
Pylon IO

① f= open(" path " , " → r
→ w

data= f.read()
f.write(" — — ")



```
# f = open("practice.txt","r")
# data = f.read()
# print(data)
# print(type(data))

# f = open("Test.txt","r")
# line1 = f.readline()
# print(line1)
#
# line2 = f.readline()
# print(line2)
#
# line3 = f.readline()
# print(line3)
#
# f.close()
```



```
# with open("Test.txt","r") as f:
#     data = f.read()
#     print(data)

with open("/Users/arunkumarsharma/Documents/pythonIO/Demo.txt","a") as f:
    f.write("\nHello Everyone")
```

Hello Everyone
We are learning Java
And Java is a easy Language

1. Replace All the Occurence of java with python
2. check weather 'learning' is present or not in the file

① f.write("   ")

② data= f.read()

newdata= data.replace("java", "python")

f.write(newdata)

③

Exception

↳ Unwanted or unexpected event occurs during the Execution of the program

4 Keyword

1. try
2. except
3. else
4. finally
5. raise

```
Try :
    // risky code  ←  Exception ✓
                        ✗
except Exception as e :
    // handled Code
```

```
else :
    // when no Exception occur

finally :
    // runs always
```

```
→ num1 = 20
→ num2 = 0
→ result = num1/num2
  print(result)
```

```
num1 = 20
num2 = 10
result = num1/num
print(result)
```

```
try :
    num1=20
    num2 = 0
    res = num1/num2
    print(res)
Except :
    print(" ____ ")
else :
    print(" Inside else")
```

Exception	Error
① It can be handled	① Can't be handled
② It occured in the program	② System level
③ types < Compile Runtime	③ ⇒ Runtime Error → less memory → less RAM