Data structure
Of
Collection

Data structure

- collection → is a single value used to store multiple value

  1- list [] → ordered and changeable , duplicated is ok

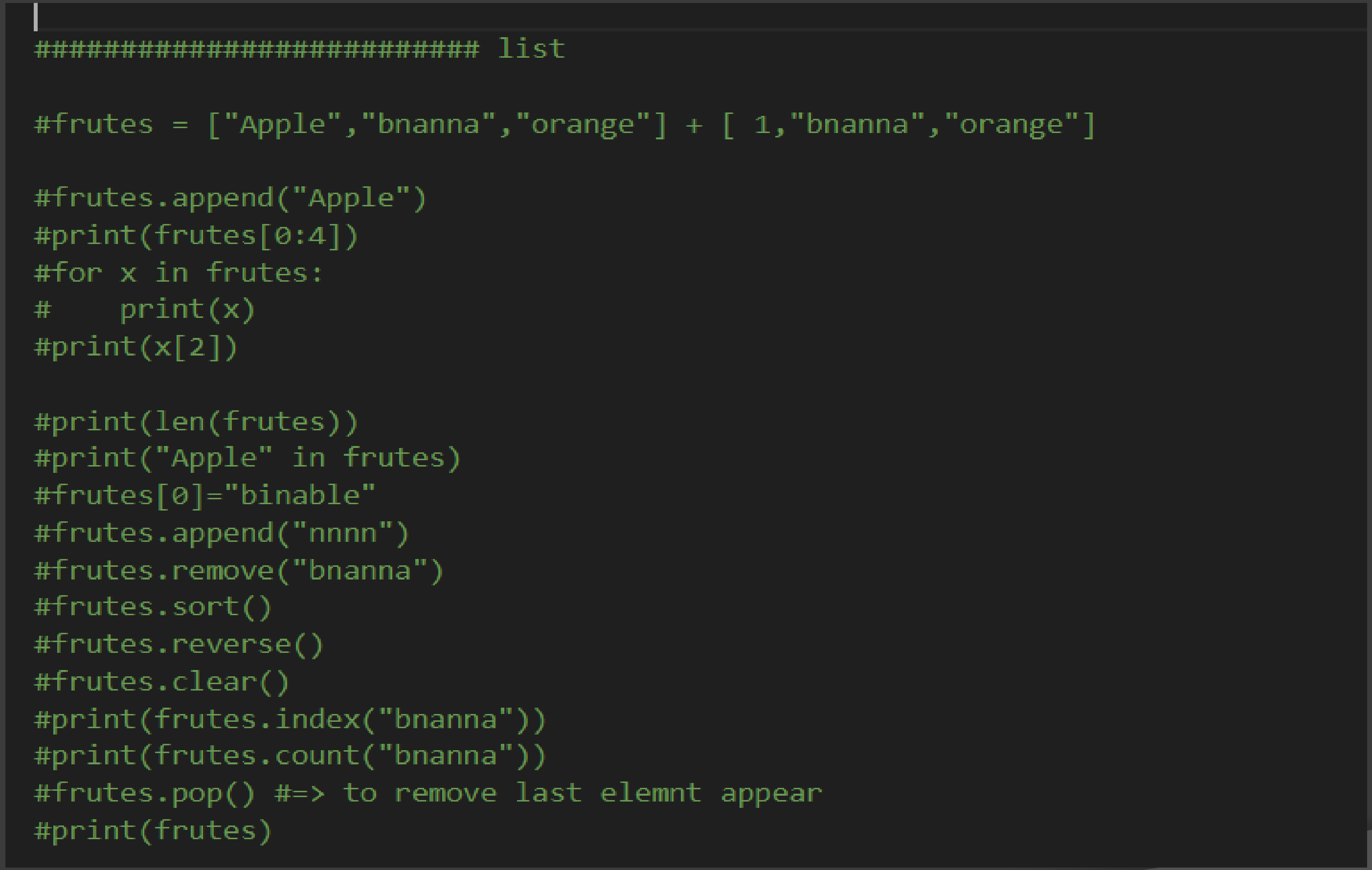
  2- set {} → unordered and immutable but add and remove ok and no duplicate

  3- Tuple () → ordered and immutable , FASTER

  4- dict {{},{}} → is a collection of {key:value} pairs
  ordered and changeable. no duplicate

1- List → ordered and changeable , duplicated is ok

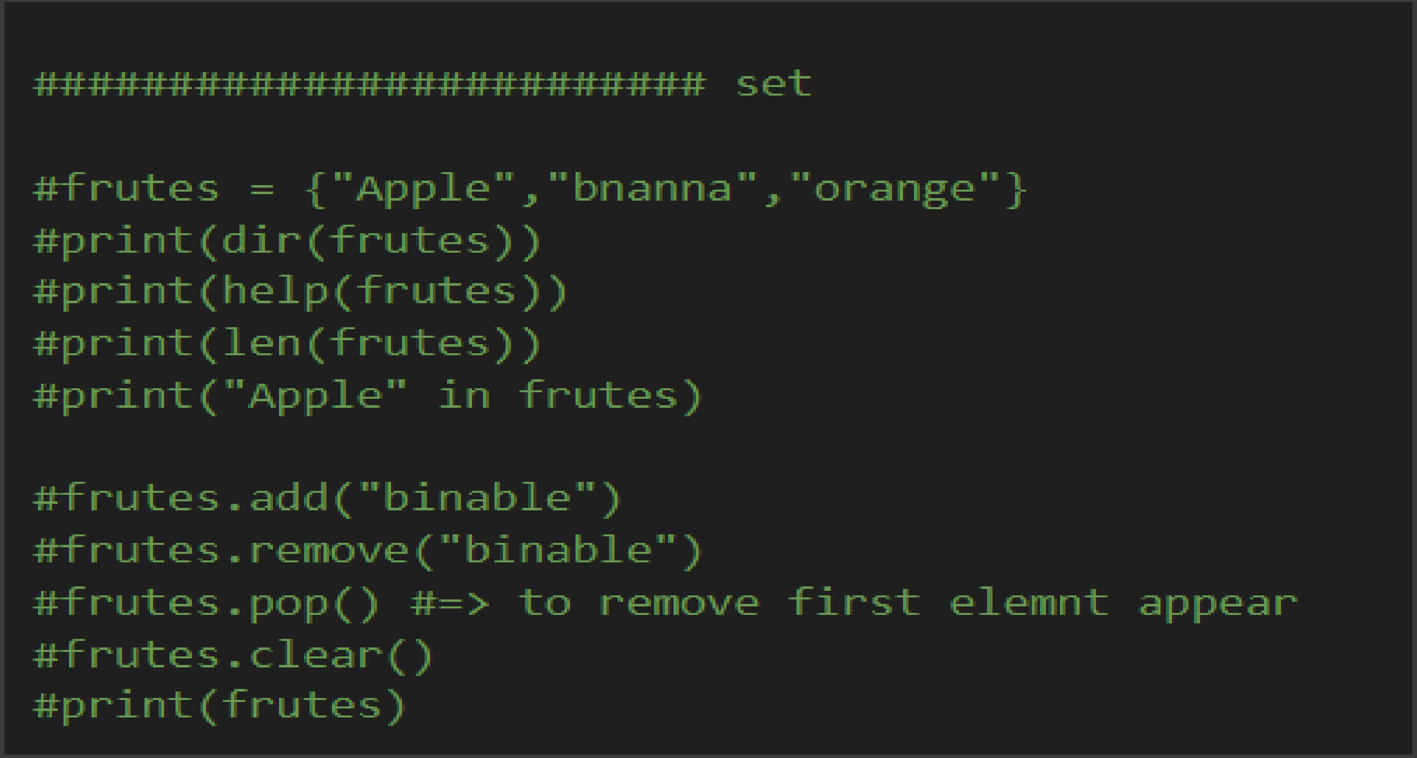
Most built in func used in list:



```
########################## list

#frutes = ["Apple","bnanna","orange"] + [ 1,"bnanna","orange"]

#frutes.append("Apple")
#print(frutes[0:4])
#for x in frutes:
#    print(x)
#print(x[2])

#print(len(frutes))
#print("Apple" in frutes)
#frutes[0]="binable"
#frutes.append("nnnn")
#frutes.remove("bnanna")
#frutes.sort()
#frutes.reverse()
#frutes.clear()
#print(frutes.index("bnanna"))
#print(frutes.count("bnanna"))
#frutes.pop() #=> to remove last elemnt appear
#print(frutes)
```

2- set {} ➔ unordered and immutable but add and remove ok and no duplicate

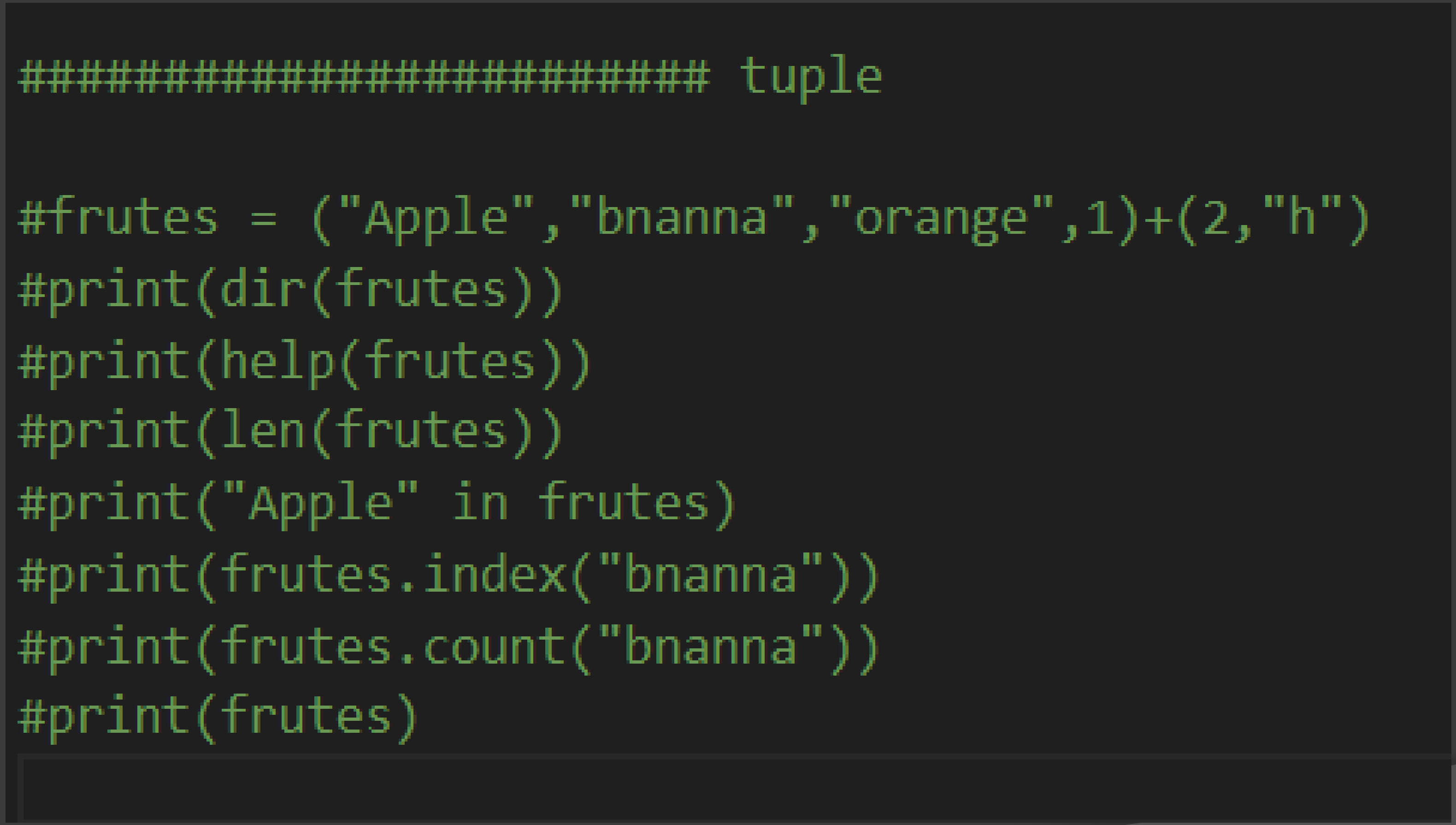
Most built in func used in set{}:



```
######################### set

#frutes = {"Apple","bnanna","orange"}
#print(dir(frutes))
#print(help(frutes))
#print(len(frutes))
#print("Apple" in frutes)

#frutes.add("binable")
#frutes.remove("binable")
#frutes.pop() #=> to remove first elemnt appear
#frutes.clear()
#print(frutes)
```

3- Tuple () ➔ ordered and immutable ,FASTER

Most built in func used in list:

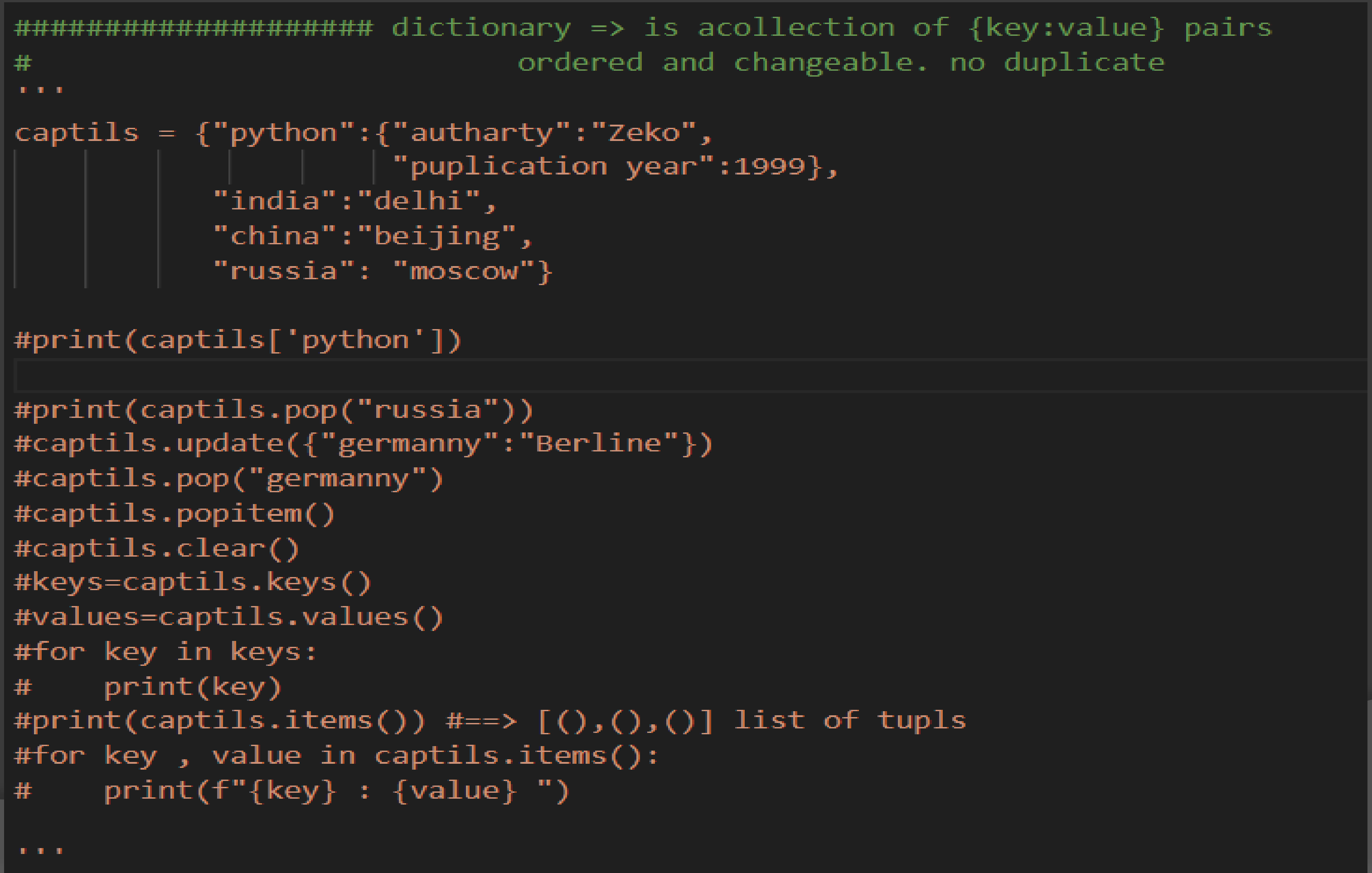


```
####################### tuple

#frutes = ("Apple","bnanna","orange",1)+(2,"h")
#print(dir(frutes))
#print(help(frutes))
#print(len(frutes))
#print("Apple" in frutes)
#print(frutes.index("bnanna"))
#print(frutes.count("bnanna"))
#print(frutes)
```

4-dictionary {{},{}} ➔ is a collection of {key : value} pairs

ordered and changeable. no duplicate

Most built in func used in dict {{},{}}:



```
#################### dictionary => is acollection of {key:value} pairs
#                           ordered and changeable. no duplicate
'''
captils = {"python":{"autharty":"Zeko",
                     "puplication year":1999},
           "india":"delhi",
           "china":"beijing",
           "russia": "moscow"}

#print(captils['python'])

#print(captils.pop("russia"))
#captils.update({"germanny":"Berline"})
#captils.pop("germanny")
#captils.popitem()
#captils.clear()
#keys=captils.keys()
#values=captils.values()
#for key in keys:
#    print(key)
#print(captils.items()) #==> [(),(),()] list of tupls
#for key , value in captils.items():
#    print(f"{key} : {value} ")

'''
```

functions

functions ➔ a block of reusable code that are designed to do one specific job. When you want to perform a particular task that you've defined in a function.

Type of func argu : default and arbitrary

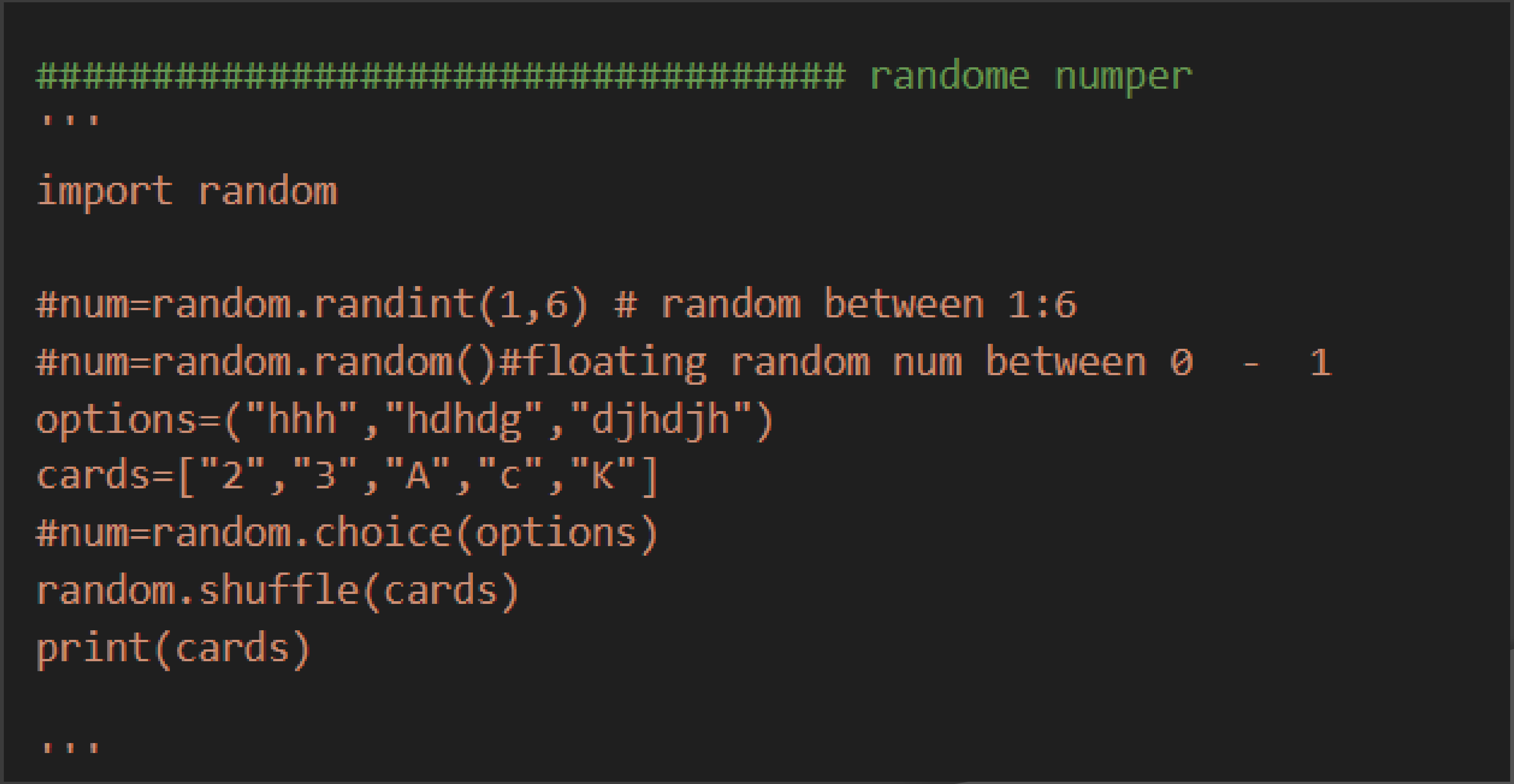
1. default ➔ used when you want to replace argu by default value to handel if user don't need it like ➔ testing(flag = 0)

2. arbitrary ➔ used when you don't know amount of parameter func needed like ➔ testing(**flag =)

Type of func with parameter when invoke : positional - keyword

1. positional → like → testing(True)
2. arbitrary → like → testing(flag = True)

Random num



```
################################### randome numper
'''
import random

#num=random.randint(1,6) # random between 1:6
#num=random.random()#floating random num between 0  -  1
options=("hhh","hdhdg","djhdjh")
cards=["2","3","A","c","K"]
#num=random.choice(options)
random.shuffle(cards)
print(cards)

'''
```

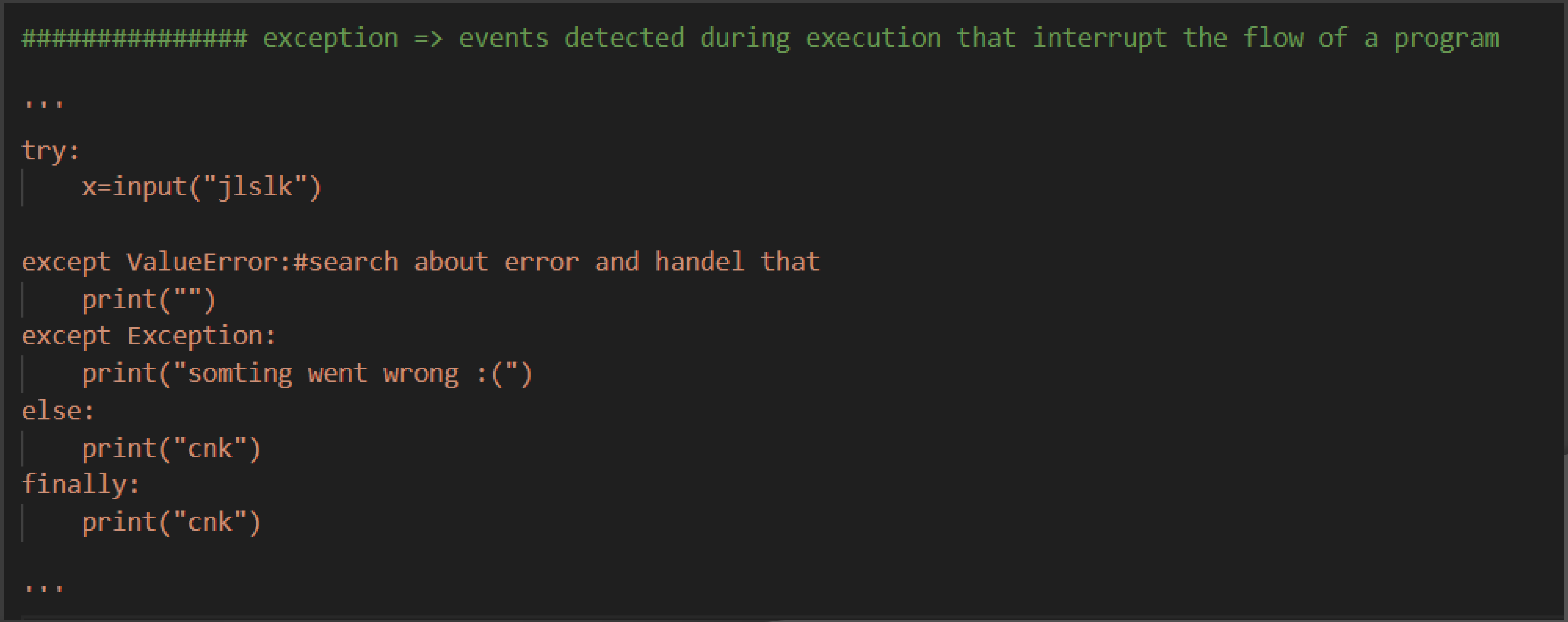
module → a file containing code you want to include in your program

→→→ use 'import' to include a module (built-in or your own)

Characteristics → useful to break up a large prog reusable separate file

handle errors

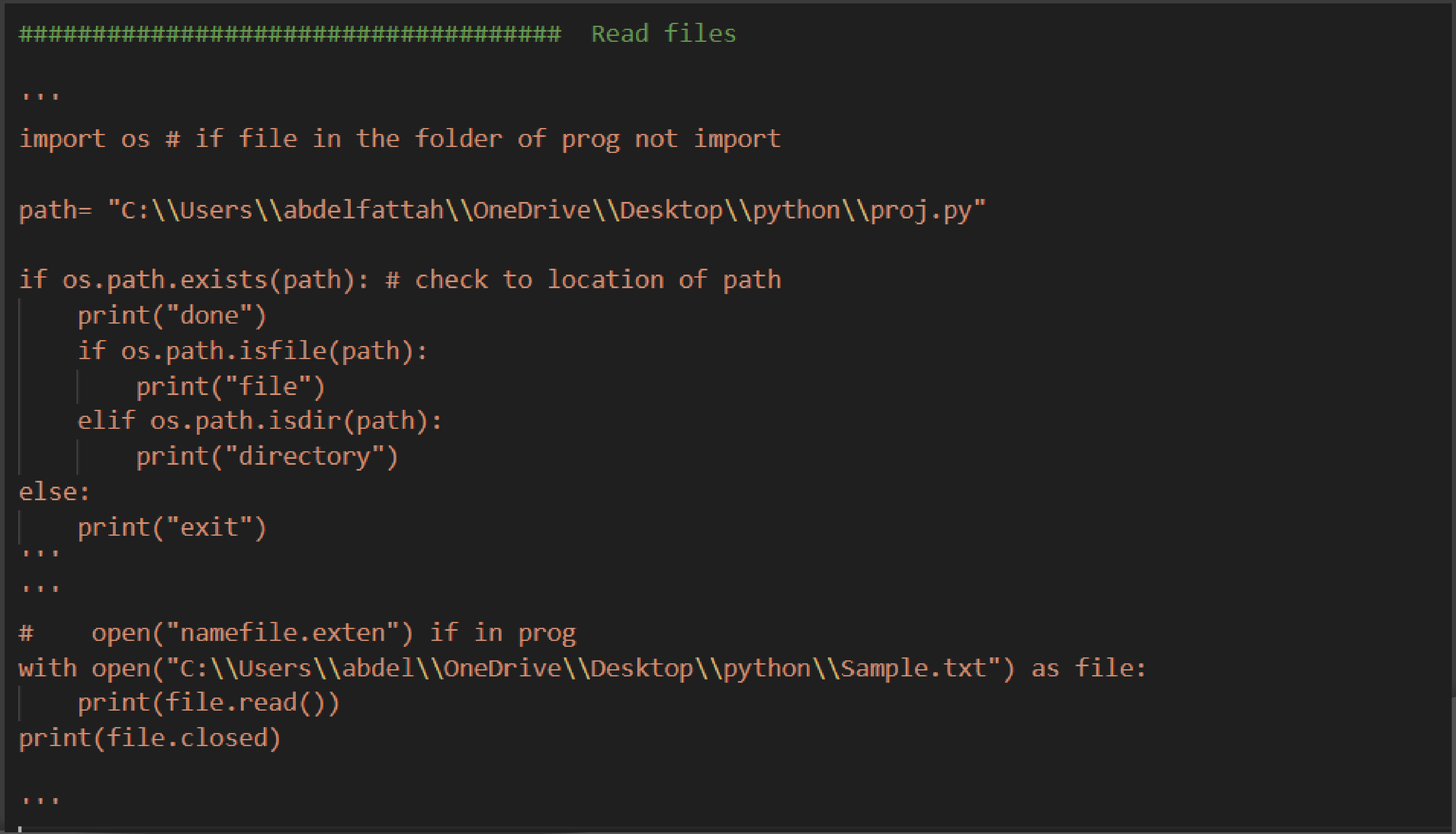
exception → events detected during execution that interrupt the flow of a program.



```
############### exception => events detected during execution that interrupt the flow of a program

'''
try:
    x=input("jlslk")

except ValueError:#search about error and handel that
    print("")
except Exception:
    print("somting went wrong :(")
else:
    print("cnk")
finally:
    print("cnk")

'''
```

Read files

- If file not in folder of prog “import os” else not important to import



```
#####################################  Read files

'''
import os # if file in the folder of prog not import

path= "C:\\Users\\abdelfattah\\OneDrive\\Desktop\\python\\proj.py"

if os.path.exists(path): # check to location of path
    print("done")
    if os.path.isfile(path):
        print("file")
    elif os.path.isdir(path):
        print("directory")
else:
    print("exit")
'''
'''
#    open("namefile.exten") if in prog
with open("C:\\Users\\abdel\\OneDrive\\Desktop\\python\\Sample.txt") as file:
    print(file.read())
print(file.closed)

'''
```

End