

```
In [1]: import platform  
platform.python_version()
```

```
Out[1]: '3.9.18'
```

```
In [2]: platform.architecture()
```

```
Out[2]: ('64bit', 'WindowsPE')
```

```
In [3]: platform.processor()
```

```
Out[3]: 'AMD64 Family 23 Model 96 Stepping 1, AuthenticAMD'
```

```
In [4]: import sys  
sys.version
```

```
Out[4]: '3.9.18 (main, Sep 11 2023, 14:09:26) [MSC v.1916 64 bit (AMD64)]'
```

```
In [5]: print(type('Hello, World!'))  
  
<class 'str'>
```

```
In [6]: 'kiio weds ankit '*2
```

```
Out[6]: 'kiio weds ankit kiio weds ankit '
```

```
In [7]: import math  
math
```

```
Out[7]: <module 'math' (built-in)>
```

```
In [8]: import my_python_module  
my_python_module
```

```
Out[8]: <module 'my_python_module' from 'C:\\Users\\ankit19.gupta\\ankit\\ankit\\ML_Code\\Python_R_Prolog_Code\\Python_Practice\\thinkpython\\my_python_module.py'>
```

```
In [9]: from my_python_module import x  
x
```

```
Out[9]: 5
```

```
In [10]: from my_python_module import print_name  
print_name
```

```
Out[10]: <function my_python_module.print_name()>
```

```
In [11]: print_name()
```

```
ankit  
Hi, ankit
```

```
In [12]: import my_python_module  
my_python_module.x
```

Out[12]: 5

```
In [13]: import math
         math.sin(math.pi/4)
```

Out[13]: 0.7071067811865476

```
In [14]: type(print_name)
```

Out[14]: function

```
In [15]: print(print_name)

<function print_name at 0x0000015A022ABEE0>
```

```
In [16]: import turtle
         bob = turtle.Turtle()
         print(bob)
         bob.fd(100)
         bob.lt(90)
         bob.fd(100)

<turtle.Turtle object at 0x0000015A022FB2E0>
```

```
In [1]: type(True)
```

Out[1]: bool

```
In [3]: not isinstance(1.7, int)
```

Out[3]: True

```
In [9]: # Estimated square root of a number "a" using Newton's Method
         a=100000
         x=1
         y=(x+a/x)/2
         print(y)
         while not(abs(y-x) < 0.001):
             x=y
             y=(x+a/x)/2
             print("estimate of sqrt(a) is", y)
```

```
50000.5
estimate of sqrt(a) is 25001.2499900001
estimate of sqrt(a) is 12502.624895005849
estimate of sqrt(a) is 6255.311607712872
estimate of sqrt(a) is 3135.6490107717154
estimate of sqrt(a) is 1583.7701676166232
estimate of sqrt(a) is 823.455321095483
estimate of sqrt(a) is 472.4474090502863
estimate of sqrt(a) is 342.0555898147884
estimate of sqrt(a) is 317.2028655357483
estimate of sqrt(a) is 316.22926477232704
estimate of sqrt(a) is 316.2277660203896
estimate of sqrt(a) is 316.2277660168379
```

```
In [11]: from math import sqrt
         sqrt(100000)
```

Out[11]: 316.22776601683796

```
In [12]: name="kateeee"
         new_name="ankit"
         for i in name:
             print(i+new_name)
```

kankit  
aankit  
tankit  
eankit  
eankit  
eankit  
eankit

```
In [13]: name="ankit"
         name.upper()
```

Out[13]: 'ANKIT'

```
In [14]: # index in a string
         name.find('k')
```

Out[14]: 2

```
In [16]: name.find('it')
```

Out[16]: 3

```
In [17]: name = 'bob'
         name.find('b', 1, 2) # finding character 'b' from index range 1 to 2 and it returns -1
```

Out[17]: -1

```
In [20]: 'nk' in 'ankit'
```

Out[20]: True

```
In [56]: fin = open('words.txt','r+') # both read and write
         fin
```

Out[56]: <\_io.TextIOWrapper name='words.txt' mode='r+' encoding='cp1252'>

```
In [57]: fin.readline()
```

Out[57]: 'ankit\n'

```
In [30]: #fin.writelines("kio")
```

```
In [59]: fin.tell()
```

Out[59]: 11

```
In [60]: fin.seek(0)
```

Out[60]: 0

```
In [61]: [line.strip() for line in fin]
```

Out[61]: ['ankit', 'kiio']

```
In [62]: fin.write("my love kiio")
```

Out[62]: 12

```
In [63]: fin.seek(0)
```

Out[63]: 0

```
In [64]: [line.strip() for line in fin]
```

Out[64]: ['ankit', 'kiiomy love kiio']

```
In [65]: fin.write("\n ankit likes kiio")
fin.seek(0)
[line.strip() for line in fin]
```

Out[65]: ['ankit', 'kiiomy love kiio', 'ankit likes kiio']

```
In [66]: with open("words.txt", 'r') as f:
print(f.readline().strip())
```

ankit

```
In [67]: ['ankit', 'likes'] + ['kiio']
```

Out[67]: ['ankit', 'likes', 'kiio']

```
In [68]: ['kiio']*3
```

Out[68]: ['kiio', 'kiio', 'kiio']

```
In [71]: l=[1,2,3]
l.extend([4,5,6])
l
```

Out[71]: [1, 2, 3, 4, 5, 6]

```
In [74]: l.sort(reverse=True)
```

```
In [75]: l
```

Out[75]: [6, 5, 4, 3, 2, 1]

```
In [78]: t = ['a', 'b', 'c']
y=t.pop()
x = t.pop(0)
t
```

Out[78]: ['b']

In [79]: y

Out[79]: 'c'

In [83]: "ankit-like-kiio".split('-')

Out[83]: ['ankit', 'like', 'kiio']

In [85]: "#".join("ankit-like-kiio".split('-'))

Out[85]: 'ankit#like#kiio'

In [86]: l=list()  
d=dict()  
s=set()  
t=tuple()  
l

Out[86]: []

In [87]: d

Out[87]: {}

In [88]: s

Out[88]: set()

In [89]: t

Out[89]: ()

In [90]: d['ankit']='kiio'  
d

Out[90]: {'ankit': 'kiio'}

In [92]: 'ankit' in d

Out[92]: True

In [94]: d.keys()

Out[94]: dict\_keys(['ankit'])

In [95]: d.values()

Out[95]: dict\_values(['kiio'])

In [97]: d.items()

Out[97]: dict\_items([('ankit', 'kiio')])

In [100... `d.get("ankit","default value if key does not exist")`

Out[100... 'kiio'

In [101... `d.get("key","default value if key does not exist")`

Out[101... 'default value if key does not exist'

In [98]: `ord('a')`

Out[98]: 97

In [99]: `ord('A')`

Out[99]: 65

In [102... 

```
verbose = True
def example1():
    if verbose:
        print('Running example1')
```

In [104... `example1()`

Running example1

In [105... 

```
been_called = False
def example2():
    been_called = True # WRONG
```

In [106... `been_called`

Out[106... False

In [108... 

```
been_called = False
def example2():
    global been_called
    been_called = True
example2()
been_called
```

Out[108... True

In [111... 

```
count = 0
def example3():
    #count=1 # now we create the local variable so no error
    count = count + 1 # WRONG because we are changing the global variable
example3()
```

-----  
**UnboundLocalError**

Traceback (most recent call last)

Cell In[111], line 5

```
2 def example3():
3     #count=1
4     count = count + 1 # WRONG
```

```
----> 5 example3()
```

```
Cell In[111], line 4, in example3()
```

```
2 def example3():
3     #count=1
----> 4     count = count + 1
```

**UnboundLocalError**: local variable 'count' referenced before assignment

```
In [115... (1,2,3) < (1,2,3)
```

```
Out[115... False
```

```
In [116... (1,2,3) < (1,3,2000000)
```

```
Out[116... True
```

```
In [118... # swap
a=1
b=2
a, b = b, a
print(a,b)
```

```
2 1
```

```
In [119... divmod(8,3)
```

```
Out[119... (2, 2)
```

```
In [120... def printall(*args):
print(args)
printall(1, 2.0, '3')
```

```
(1, 2.0, '3')
```

```
In [121... divmod(*[8,3])
```

```
Out[121... (2, 2)
```

```
In [122... printall(*[1,2.0,'3'])
```

```
(1, 2.0, '3')
```

```
In [127... s = 'abc'
t = [0, 1, 2]
zip(s, t)
```

```
Out[127... <zip at 0x29063fb42c0>
```

```
In [129... list(zip(s,t))
```

```
Out[129... [('a', 0), ('b', 1), ('c', 2)]
```

```
In [128... list(zip('Anne', 'Elk'))
```

```
Out[128... [('A', 'E'), ('n', 'l'), ('n', 'k')]
```

```
In [130... def has_match(t1, t2):  
    for x, y in zip(t1, t2):  
        if x == y:  
            return True  
    return False
```

```
In [131... t1=['ankit','kioo']  
t2=['ankit','kioo']  
has_match(t1, t2)
```

Out[131... True

```
In [132... for key, value in d.items():  
    print(key, value)
```

ankit kiio

```
In [133... d.items()
```

Out[133... dict\_items([('ankit', 'kiio')])

```
In [134... d = dict(zip('abc', range(3)))  
d
```

Out[134... {'a': 0, 'b': 1, 'c': 2}

```
In [135... import string  
string.punctuation
```

Out[135... '!"#\$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~'

```
In [136... import random  
random.randint(5, 10)
```

Out[136... 10

```
In [137... t = [1, 2, 3]  
random.choice(t)
```

Out[137... 3

```
In [1]: import os  
os.path.abspath('output.txt')
```

Out[1]: 'C:\\Users\\ankit19.gupta\\ankit\\ankit\\ML\_Code\\Python\_R\_Prolog\_Code\\Python\_Practice\\thinkpython\\output.txt'

```
In [5]: os.path.exists('output.txt')
```

Out[5]: True

```
In [6]: os.path.isdir('output.txt')
```

Out[6]: False



```
In [7]: os.path.isfile('output.txt')
```

```
Out[7]: True
```

```
In [9]: os.listdir(os.getcwd())
```

```
Out[9]: ['.git',
          '.ipynb_checkpoints',
          '158-0.txt',
          'captions.bak',
          'captions.dat',
          'captions.dir',
          'Chapter_10_Lists.ipynb',
          'Chapter_10_Lists.pdf',
          'Chapter_11_Dictionary.ipynb',
          'Chapter_11_Dictionary.pdf',
          'Chapter_12_Tuples.ipynb',
          'Chapter_12_Tuples.pdf',
          'Chapter_13_Files.ipynb',
          'Chapter_13_Files.pdf',
          'Chapter_14_Classes_and_objects.ipynb',
          'Chapter_14_Classes_and_objects.pdf',
          'Chapter_15_Classes_and_functions.ipynb',
          'Chapter_15_Classes_and_functions.pdf',
          'Chapter_16_Classes_and_methods.ipynb',
          'Chapter_16_Classes_and_methods.pdf',
          'Chapter_17_Inheritance.ipynb',
          'Chapter_17_Inheritance.pdf',
          'Chapter_1_The_way_of_the_program.ipynb',
          'Chapter_1_The_way_of_the_program.pdf',
          'Chapter_2_Variables_expressions_and_statements.ipynb',
          'Chapter_2_Variables_expressions_and_statements.pdf',
          'Chapter_3_Functions.ipynb',
          'Chapter_3_Functions.pdf',
          'Chapter_4_Case_study_interface_design.ipynb',
          'Chapter_4_Case_study_interface_design.pdf',
          'Chapter_5_Conditionals_and_recursion.ipynb',
          'Chapter_5_Conditionals_and_recursion.pdf',
          'Chapter_6_Fruitful_functions.ipynb',
          'Chapter_6_Fruitful_functions.pdf',
          'Chapter_7_Iteration.ipynb',
          'Chapter_7_Iteration.pdf',
          'Chapter_8_Strings.ipynb',
          'Chapter_8_Strings.pdf',
          'Chapter_9_Case_Study.ipynb',
          'Chapter_9_Case_Study.pdf',
          'mypolygon.py',
          'my_python_module.py',
          'output.txt',
          'Summary.ipynb',
          'Summary.txt',
          'wc.py',
          'words.txt',
          '__pycache__']
```

```
In [12]: os.path.isfile(os.path.join('C:\\Users\\ankit19.gupta\\ankit\\ankit\\ML_Code\\Python_R_
```

```
Out[12]: True
```

```
In [13]: import dbm
          db = dbm.open('captions', 'c')
```

```
In [14]: db['ankit']='kiio'
```

```
In [15]: db['ankit']
```

```
Out[15]: b'kiio'
```

```
In [16]: for key in db.keys():
          print(key, db[key])
```

```
b'cleese.png' b'Photo of John Cleese doing a silly walk.'
b'ankit' b'kiio'
```

```
In [17]: db.close()
```

```
In [21]: import pickle
          t = [1, 2, 3]
          s=pickle.dumps(t)
```

```
In [22]: t2 = pickle.loads(s)
          t2
```

```
Out[22]: [1, 2, 3]
```

```
In [26]: os.popen('ls -l').read()
```

```
Out[26]: 'total 7446\n-rw-r--r-- 1 Ankit19.Gupta 1049089 927445 Dec 25 01:10 158-0.txt\n-rw-r--r--
-- 1 Ankit19.Gupta 1049089 19909 Mar 6 13:52 Chapter_10_Lists.ipynb\n-rw-r--r-- 1 Ank
it19.Gupta 1049089 44071 Mar 6 13:54 Chapter_10_Lists.pdf\n-rw-r--r-- 1 Ankit19.Gupta
1049089 31874 Apr 15 01:15 Chapter_11_Dictionary.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 104
9089 56848 Mar 6 15:13 Chapter_11_Dictionary.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089
34636 Mar 6 16:05 Chapter_12_Tuples.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 57912 M
ar 6 16:06 Chapter_12_Tuples.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 40540 Apr 15 11:
10 Chapter_13_Files.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 419125 Mar 6 17:07 Chapt
er_13_Files.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 39704 Mar 7 11:18 Chapter_14_Clas
ses_and_objects.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 326288 Mar 7 11:25 Chapter_1
4_Classes_and_objects.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 11364 Mar 7 11:25 Chapt
er_15_Classes_and_functions.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 38935 Mar 7 11:
45 Chapter_15_Classes_and_functions.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 31660 Mar
7 15:10 Chapter_16_Classes_and_methods.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 582632
Mar 7 15:14 Chapter_16_Classes_and_methods.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 92
794 Mar 7 16:32 Chapter_17_Inheritance.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 94377
0 Mar 7 16:35 Chapter_17_Inheritance.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 7395 Ap
r 14 19:06 Chapter_1_The_way_of_the_program.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089
32432 Mar 6 00:53 Chapter_1_The_way_of_the_program.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049
089 13638 Apr 14 19:37 Chapter_2_Variables_expressions_and_statements.ipynb\n-rw-r--r-
- 1 Ankit19.Gupta 1049089 44328 Mar 6 01:23 Chapter_2_Variables_expressions_and_state
ments.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 18491 Apr 14 21:04 Chapter_3_Functions.i
pynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 44175 Mar 6 10:52 Chapter_3_Functions.pdf\n-
rw-r--r-- 1 Ankit19.Gupta 1049089 6359 Mar 6 11:10 Chapter_4_Case_study_interface_de
sign.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 31557 Mar 6 11:18 Chapter_4_Case_study
_interface_design.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 6636 Mar 6 11:37 Chapter_5
_Conditionals_and_recursion.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 30574 Mar 6 1
1:39 Chapter_5_Conditionals_and_recursion.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 30
35 Mar 6 11:42 Chapter_6_Fruitful_functions.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089
23841 Mar 6 11:43 Chapter_6_Fruitful_functions.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089
4318 Mar 6 11:54 Chapter_7_Iteration.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 28156
Mar 6 11:55 Chapter_7_Iteration.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 11888 Mar 6
12:09 Chapter_8_Strings.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 36587 Mar 6 12:09
Chapter_8_Strings.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 2281243 Mar 6 13:25 Chapter_
9_Case_Study.ipynb\n-rw-r--r-- 1 Ankit19.Gupta 1049089 1178419 Mar 6 13:30 Chapter_9_C
```

```

ase_Study.pdf\n-rw-r--r-- 1 Ankit19.Gupta 1049089 35976 Apr 15 11:17 Summary.ipynb\n-
rw-r--r-- 1 Ankit19.Gupta 1049089 19801 Apr 15 11:17 Summary.txt\ndrwxr-xr-x 1 Ankit1
9.Gupta 1049089 0 Apr 14 20:01 __pycache__\n-rw-r--r-- 1 Ankit19.Gupta 1049089
42 Apr 15 10:58 captions.bak\n-rw-r--r-- 1 Ankit19.Gupta 1049089 516 Apr 15 10:58 c
aptions.dat\n-rw-r--r-- 1 Ankit19.Gupta 1049089 42 Apr 15 11:00 captions.dir\n-rw-
r--r-- 1 Ankit19.Gupta 1049089 61 Apr 14 20:00 my_python_module.py\n-rw-r--r-- 1 A
nkit19.Gupta 1049089 2428 Nov 11 18:10 mypolygon.py\n-rw-r--r-- 1 Ankit19.Gupta 1049
089 2 Dec 12 13:34 output.txt\n-rw-r--r-- 1 Ankit19.Gupta 1049089 132 Dec 12
13:35 wc.py\n-rw-r--r-- 1 Ankit19.Gupta 1049089 42 Apr 15 02:35 words.txt\n'

```

```

In [27]: class Point:
         '''
         represents point in a 2d
         '''

```

```

In [28]: print(Point)

<class '__main__.Point'>

```

```

In [29]: blank=Point()
         print(blank)

<__main__.Point object at 0x000001E27C74C5E0>

```

```

In [30]: blank.x = 3.0
         blank.y = 4.0

```

```

In [31]: '(%g, %g)' % (blank.x, blank.y)

```

```

Out[31]: '(3, 4)'

```

```

In [32]: isinstance(blank,Point)

```

```

Out[32]: True

```

```

In [34]: hasattr(blank,'x')

```

```

Out[34]: True

```

```

In [35]: class Point:
         """Represents a point in 2-D space."""
         p = Point()
         p.x = 3
         p.y = 4
         vars(p)

```

```

Out[35]: {'x': 3, 'y': 4}

```

```

In [ ]:

```