

Different types of importing, Packing and Unpacking of iterables in Python

February 6, 2023

1 import module_name

```
[ ]: import math
     l = math.lcm(12, 18)
     print(l)
```

```
[2]: lcm(12, 18)
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\923456516.py in <cell line: 1>()
----> 1 lcm(12, 18)

NameError: name 'lcm' is not defined
```

```
[3]: import math
     print(math.pi)
```

3.141592653589793

```
[4]: import math
     print(pi)
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\2866052711.py in <cell line: 2>()
      1 import math
----> 2 print(pi)

NameError: name 'pi' is not defined
```

2 import module_name as alias_name

- import math as mt

- import random as rd

```
[5]: import random as rd
      print(random.randint(1, 100))
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\1095326628.py in <cell line: 2>()
      1 import random as rd
----> 2 print(random.randint(1, 100))

NameError: name 'random' is not defined
```

```
[6]: import random as rd
      print(rd.randint(1, 100))
```

8

3 import m1, m2, m3....

```
[7]: import math, random, string
      print(math.sqrt(25))
      print(random.randint(1, 100))
      print(string.ascii_lowercase)
```

5.0

48

abcdefghijklmnopqrstuvwxyz

4 import m1 as a1, m2 as a2, ...

```
[8]: import math as mt, random as rd, string as st
      print(mt.sqrt(25))
      print(rd.randint(1, 100))
      print(st.ascii_lowercase)
```

5.0

96

abcdefghijklmnopqrstuvwxyz

5 from module_name import function_name

```
[9]: from math import factorial
      print(factorial(5))
```

120

6 from module_name import f1, f2, f3....

```
[10]: from math import gcd, lcm, factorial
      print(gcd(12, 18))
      print(lcm(12, 18))
      print(factorial(6))
```

```
6
36
720
```

```
[11]: from math import factorial as f
      print(f(int(input())))
```

```
5
120
```

```
[12]: from math import * # import everything from the module
      print(comb(5, 2))
      print(perm(5, 2))
      print(lcm(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
```

```
10
20
2520
```

7 eval()

- `eval("x**2 + x + 1", {'x': 5}) -> 31`

```
[13]: output = eval("x**2 + x + 1", {'x': 5})
      print(output)
```

```
31
```

```
[14]: lst = list(map(int, input().split()))
      print(lst)
```

```
10 20 30
[10, 20, 30]
```

```
[15]: x = int(input())
      print(x)
```

```
10
10
```

```
[17]: x = eval(input())
      print(x, type(x))
```

```
10
10 <class 'int'>
```

```
[18]: x = eval(input())
      print(x, type(x))
```

```
12.2
12.2 <class 'float'>
```

```
[19]: x = eval(input())
      print(x, type(x))
```

```
hello
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\3324904713.py in <cell line: 1>()
----> 1 x = eval(input())
      2 print(x, type(x))

<string> in <module>

NameError: name 'hello' is not defined
```

```
[20]: x = hello
      print(x, type(x))
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\2376646085.py in <cell line: 1>()
----> 1 x = hello
      2 print(x, type(x))

NameError: name 'hello' is not defined
```

```
[22]: x = eval("'hello'")
      print(x)
```

```
hello
```

```
[23]: x = 'hello'
      print(x)
```

```
hello
```

```
[25]: print('hello')
```

```
hello
```

```
[ ]: string = "y**2 + z - 5"
```

```
[27]: y = 5  
z = 4  
res = y**2 + z - 5  
print(res)
```

24

```
[28]: p, q = 7, 8  
string = "p**2 + q - 5"  
res = eval(string)  
print(res)
```

```
-----  
NameError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_27728\600847751.py in <cell line: 2>()  
      1 string = "p**2 + q - 5"  
----> 2 res = eval(string)  
      3 print(res)  
  
<string> in <module>  
  
NameError: name 'p' is not defined
```

```
[31]: p, q = 7, 8  
string = "p**2 + q - 5"  
res = eval(string)  
print(res)
```

52

```
[30]: k = 2  
d = 3  
res = k ** 2 + d - 5  
print(res)
```

2

```
[32]: string = '[10, 20, 30]'  
print(string[0])
```

[

```
[33]: string = '[10, 20, 30]'  
lst = eval(string)  
print(lst[0])
```

10

```
[34]: x, y, z = map(int, input().split())
      expression = "x ** 3 + y ** 2 + z"
      print(eval(expression))
```

```
10 5 4
1029
```

```
[36]: expression = "x ** 3 + y ** 2 + z"
      print(eval(expression, {'x': int(input()), 'y': int(input()), 'z':
                               ↪int(input())}))
```

```
10
5
4
1029
```

8 packing, unpacking

```
[37]: lst = [10, 20, 30, 40] # packing
      # unpacking operator is *
      print(*lst)
```

```
10 20 30 40
```

```
[38]: print(10, 20, 30, 40)
```

```
10 20 30 40
```

```
[39]: import math
      # gcd, lcm
      print(math.lcm(12, 18))
```

```
36
```

```
[40]: import math
      # gcd, lcm
      print(math.lcm(1, 2, 3))
```

```
6
```

```
[41]: import math
      # gcd, lcm
      print(math.lcm(1, 2, 3, 4))
```

```
12
```

```
[42]: lst = [1, 2, 3, 4]
      print(math.lcm(lst))
```

```

-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27728\1960550506.py in <cell line: 2>()
      1 lst = [1, 2, 3, 4]
----> 2 print(math.lcm(lst))

TypeError: 'list' object cannot be interpreted as an integer

```

```
[43]: lst = [1, 2, 3, 4]
      print(*lst)
```

1 2 3 4

```
[46]: import math
      # gcd, lcm
      lst = list(range(1, 11))
      # print(lst)
      print(math.lcm(*lst))
```

2520

```
[47]: import math
      # gcd, lcm
      lst = list(range(1, 11))
      # print(lst)
      print(math.gcd(*lst))
```

1

```
[53]: employees = [("Pavan", 1234, 2020, 50000),
                  ("Kiran", 4321, 2021, 75000),
                  ("Rajesh", 7895, 2019, 100000),
                  ("Sudhir", 5479, 2017, 125000)]
      # Employee_name with emp_id joined in year_of_join is earning salary currently
      for i in employees:
          print("{} with {} joined in {} is earning {}".format(*i))
```

Pavan with 1234 joined in 2020 is earning 50000
 Kiran with 4321 joined in 2021 is earning 75000
 Rajesh with 7895 joined in 2019 is earning 100000
 Sudhir with 5479 joined in 2017 is earning 125000

```
[ ]:
```

```
[54]: def location(city, country, continent):
      print(f'{city} is in {country} which is in {continent}')

      location("Hyderabad", "India", "Asia")
```

Hyderabad is in India which is in Asia

```
[55]: def location(city, country, continent):  
      print(f'{city} is in {country} which is in {continent}')
```

location("India", "Hyderabad", "Asia")

India is in Hyderabad which is in Asia

```
[56]: def location(city, country, continent):  
      print(f'{city} is in {country} which is in {continent}')
```

location(country="India", city="Hyderabad", continent="Asia") # *keywords*

Hyderabad is in India which is in Asia

```
[57]: def location(city, country, continent):  
      print(f'{city} is in {country} which is in {continent}')
```

location('Hyderabad', 'India')

```
-----  
TypeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_27728\417288592.py in <cell line: 5>()  
      3  
      4  
----> 5 location('Hyderabad', 'India')
```

TypeError: location() missing 1 required positional argument: 'continent'

```
[58]: def location(city, country, continent):  
      print(f'{city} is in {country} which is in {continent}')
```

location('Hyderabad', 'India', 'Asia', 'Earth')

```
-----  
TypeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_27728\2367046582.py in <cell line: 5>()  
      3  
      4  
----> 5 location('Hyderabad', 'India', 'Asia', 'Earth')
```

TypeError: location() takes 3 positional arguments but 4 were given

```
[ ]: # default arguments
```



```
[59]: lst = [10, 20, 30]
      sum(lst) #
```

[59]: 60

```
[60]: lst = [10, 20, 30]
      sum(lst, 10) #
```

[60]: 70

```
[ ]: sum(iterable, start=0)
```

```
[61]: lst = [10, 20, 30]
      s = 0
      for i in lst:
          s += i
      print(s)
```

60

```
[62]: def location(city, country, continent='Asia'):
      print(f'{city} is in {country} which is in {continent}')

      location('Hyderabad', 'India')
```

Hyderabad is in India which is in Asia

```
[64]: def location(city, country, continent='Asia'):
      print(f'{city} is in {country} which is in {continent}')

      location('Berlin', 'Germany', 'Europe')
```

Berlin is in Germany which is in Europe