(5.1) For on Itearable, generators or iterators (range, enumerate, zip)

1. Iterable

In [4]:

```
for digit in 12345:
    print(digit ** 2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_10904/3732127976.py in <module>
----> 1 for digit in 12345:
      2     print(digit ** 2)

TypeError: 'int' object is not iterable
```

In [5]:

```
for digit in '12345':
    print(digit ** 2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_10904/1523694664.py in <module>
      1 for digit in '12345':
----> 2     print(digit ** 2)

TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
```

In [7]:

```
for digit in '12345':
    print(digit * int(digit))
```

```
1
22
333
4444
55555
```

str.ljust(20) --> {str:<20}

str.rjust(20) --> {str:>20}

str.center(20) --> {str:^20}

In [8]:

```
d = { 'name': 'python',
    'website': 'www.python.org',
    'father': 'guido van rossum',
    'year': 1994,
    'type': 'programming langauge',
     'domain': 'general purpose language'
    }

for key in d:
    value = d[key]
    print(f"{key:>20} = {value}")
```

```
                name = python
             website = www.python.org
              father = guido van rossum
```

```
                year = 1994
                type = programming langauge
              domain = general purpose language
```

In [ ]:

```

```

In [9]:

```
d = { 'name': 'python',
    'website': 'www.python.org',
    'father': 'guido van rossum',
    'year': 1994,
    'type': 'programming langauge',
     'domain': 'general purpose language'
    }
for item in d.items():
    print(item)
```

```
('name', 'python')
('website', 'www.python.org')
('father', 'guido van rossum')
('year', 1994)
('type', 'programming langauge')
('domain', 'general purpose language')
```

In [ ]:

```

```

In [13]:

```
d = { 'name': 'python',
    'website': 'www.python.org',
    'father': 'guido van rossum',
    'year': 1994,
    'type': 'programming langauge',
     'domain': 'general purpose language'
    }
for item in d.items():
    key, value = item
    print(f"{key:>20} = {value}")
```

```
                name = python
             website = www.python.org
              father = guido van rossum
                year = 1994
                type = programming langauge
              domain = general purpose language
```

In [ ]:

```

```

In [14]:

```
d = { 'name': 'python',
    'website': 'www.python.org',
    'father': 'guido van rossum',
    'year': 1994,
    'type': 'programming langauge',
     'domain': 'general purpose language'
    }
for (key, value) in d.items():
    print(f"{key:>20} = {value}")
```

```
                name = python
```

```
website = www.python.org
 father = guido van rossum
   year = 1994
   type = programming langauge
 domain = general purpose language
```

In [ ]:

In [2]:

```
info = [
    ['sachin', 'Developer', 'Jaipur'],
    ['rajat', 'Admin', 'Ganganagar'],
    ['yadvendra', 'Devops', 'Jhunjhunu'],
    ['simran', 'developer', 'jaipur'],
]
for (name, role, city) in info:
    print("-"*34)
    print(f"|{name:^10}|{role:^10}|{city:^10}|")
else:
    print("-"*34)
```

```
----------------------------------
|  sachin  |Developer |  Jaipur  |
----------------------------------
|  rajat   |  Admin   |Ganganagar|
----------------------------------
|yadvendra |  Devops  |Jhunjhunu |
----------------------------------
|  simran  |developer |  jaipur  |
----------------------------------
```

In [16]:

```
for item in info:
    print(item)
    break
```

```
['sachin', 'Developer', 'Jaipur']
```

In [17]:

```
sells = [
    ['apple', 20, 50000],
    ['samsung', 30, 40000],
    ['oppo', 50, 15000]
]

total_sum = 0

print("-"*43)
print(f"|{'Product':^20}|{'total sell':^20}|")
for (product, quant, cost) in sells:
    total_cost = quant * cost
    total_sum += total_cost
    print('-'*43)
    print(f"|{product:^20}|{total_cost:^20}|")
else:
    print("-"*43)
    print(f"|{'total':^20}|{total_sum:^20}|")
    print("-"*43)
```

```
-------------------------------------------
|      Product       |     total sell     |
-------------------------------------------
|       apple        |      1000000       |
-------------------------------------------
|      samsung       |      1200000       |
-------------------------------------------
|         oppo          |       750000      |
-------------------------------------------
|        total          |      2950000      |
-------------------------------------------
```

2. generators or iterators

```
special objects in python used to generate sequences

lazy execution or eager execution

    1. range
    2. enumerate
    3. zip
```

A. range

In [24]:

```python
a = range(5)
print(a)
```

```
range(0, 5)
```

In [25]:

```python
a = range(5)# start =0, end=5, step=1
print(*a)
```

```
0 1 2 3 4
```

In [26]:

```python
a = range(1, 10)# start=1, end=10, step=1
print(*a)
```

```
1 2 3 4 5 6 7 8 9
```

In [27]:

```python
a = range(10, 1, 2)
print(*a)
```

In [28]:

```python
a = range(10, 1, -2)
print(*a)
```

```
10 8 6 4 2
```

In [29]:

```python
n = int(input())
print(*range(n, n*11, n), sep='\n')
```

```
5
5
10
15
20
25
30
35
40
45
50
```

In [19]:

```
lst = [34, 64, 35, 77, 34,  31, 12, 34, 10]
key = int(input())
flag = False
for i in range(len(lst)):
    item = lst[i] # O(1)
    if item == key:
        flag = True
        print(f"found at index {i}")
else:
    if not flag:
        print("not found")
```

```
34
found at index 0
found at index 4
found at index 7
```

In [3]:

```
for i in range(1, 11):
    print(("*"*(2*i-1)).center(21, '-'))
```

```
----------*----------
---------***---------
--------*****--------
-------*******-------
------*********------
-----***********-----
----*************----
---***************---
--*****************--
-*******************-
```

In [4]:

```
for i in range(1, 11):
    print(("*"*i).rjust(10))
```

```
         *
        **
       ***
      ****
     *****
    ******
   *******
  ********
 *********
**********
```

In [5]:

```
for i in range(1, 11):
    print((f"{chr(96+i)}"*(2*i-1)).center(21, '-'))
```

```
----------a----------
---------bbb---------
--------ccccc--------
-------ddddddd-------
------eeeeeeeee------
-----fffffffffff-----
----ggggggggggggg----
---hhhhhhhhhhhhhhh---
--iiiiiiiiiiiiiiiii--
-jjjjjjjjjjjjjjjjjjj-
```

In [7]:

```
for i in range(1,11):
    print(("_._"*(2*i-1)).center(63,'-'))
```

```
------------------------------_._------------------------------
```

```
---------------------------_.__.__._---------------------------
------------------------_.__.__.__.__._------------------------
---------------------_.__.__.__.__.__.__._---------------------
------------------_.__.__.__.__.__.__.__.__._------------------
---------------_.__.__.__.__.__.__.__.__.__.__._---------------
------------_.__.__.__.__.__.__.__.__.__.__.__.__._------------
---------_.__.__.__.__.__.__.__.__.__.__.__.__.__.__._---------
------_.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__._------
---_.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__._---
```

B. enumerate

In [9]:

```
print(*enumerate('hello'), sep='\n')
```

```
(0, 'h')
(1, 'e')
(2, 'l')
(3, 'l')
(4, 'o')
```

In [ ]:

In [8]:

```
lst = [ 'java', 'c', 'c++', 'ruby', 'python']
c = 1
for item in lst:
    print(f"{c}.{item}")
    c += 1
```

```
1.java
2.c
3.c++
4.ruby
5.python
```

In [10]:

```
lst = [ 'java', 'c', 'c++', 'ruby', 'python']

print(*enumerate(lst,start=1),sep='\n')
```

```
(1, 'java')
(2, 'c')
(3, 'c++')
(4, 'ruby')
(5, 'python')
```

In [14]:

```
lst = [ 'java', 'c', 'c++', 'ruby', 'python']

for c,item in enumerate(lst,start=1):
    print(f"{c}. {item}")
```

```
1. java
2. c
3. c++
4. ruby
5. python
```

In [ ]:

In [15]:

```
names = [ 'sachin', 'rajat' ,'gaurav',
         'akhilesh', 'ravi', 'simran', 'nidhi',
         'shivani', 'yadvendra']
```

In [16]:

```
item = input('name: ').strip().lower()

if item in names:
    i = names.index(item)
    print(f"{item} fount at loc {i}")
else:
    print(f"{item} is not in list")
```

```
name: pankaj
pankaj is not in list
```

In [18]:

```
names = [ 'sachin', 'rajat' ,'gaurav',
         'akhilesh', 'ravi', 'simran', 'nidhi',
         'shivani', 'yadvendra']

key = input("name: ").strip().lower()
for i, item in enumerate(names):
    if item == key:
        print(f"Item Found at loc {i}")
        break
else:
    print("Item Not Found")
```

```
name: ravi
Item Found at loc 4
```

C. zip

In [1]:

```
s1 = 'aeoe'
s2 = 'wsm'
for c1,c2 in zip(s1, s2):
    print(c1,c2 , sep='',end='')
```

```
awesom
```

In [ ]:

In [21]:

```
s1 = 'aeoe'
s2 = 'wsm '
for c1,c2 in zip(s1, s2):
    print(c1,c2 , sep='',end='')
```

```
awesome
```

In [ ]:

In [23]:

```
one = [ 'java', 'c', 'c++', 'python']
two = [ 'is complicated.', 'is boring.','is good', 'is awesome.']

for a,b in zip(one,two):
    print(a,b)
```

```
java is complicated.
c is boring.
```

```
c++ is good
python is awesome.
```

In [ ]:

```

```

In [20]:

```python
product  = ['A', 'B', 'C', 'D', 'E', 'F']
price    = [30, 60, 80, 34, 100, 57]
quantity = [5, 6, 8, 3, 4, 6]

print('-'*23)
print(f"|{'Product':^10}|{'Sell':^10}|")
print('-'*23)

for i in range(len(product)):
    pd = product[i]
    pr = price[i]
    q  = quantity[i]
    s  = pr*q
    print(f"|{pd:^10}|{s:^10}|")
    print('-'*23)
```

```
-----------------------
| Product  |   Sell   |
-----------------------
|    A     |   150    |
-----------------------
|    B     |   360    |
-----------------------
|    C     |   640    |
-----------------------
|    D     |   102    |
-----------------------
|    E     |   400    |
-----------------------
|    F     |   342    |
-----------------------
```

In [26]:

```python
product  = ['A', 'B', 'C', 'D', 'E', 'F']
price    = [30, 60, 80, 34, 100, 57]
quantity = [5, 6, 8, 3, 4, 6]

print('-'*23)
print(f"|{'Product':^10}|{'Sell':^10}|")
print('-'*23)

for pd, pr, q in zip(product, price, quantity):
    s  = pr*q
    print(f"|{pd:^10}|{s:^10}|")
    print('-'*23)
```

```
-----------------------
| Product  |   Sell   |
-----------------------
|    A     |   150    |
-----------------------
|    B     |   360    |
-----------------------
|    C     |   640    |
-----------------------
|    D     |   102    |
-----------------------
|    E     |   400    |
-----------------------
```

```
|    F      |   342    |
------------------------
```

In [ ]:

```

```