```
1  python class 2020
2  coder: Mohammad Foroughi(+989226219173)
```

Looping in python

```
1  for elements in sequence_type:
2      instruction
```

In [2]:
```python
1  a = [0, 1, 2, 3, 4]
2  '''
3  a[0] = 0
4  a[1] = 1
5  .
6  .
7  .
8  a[4] = 4
9  '''
10
11  for element in a:
12      print(element)
13
14  print("\n")
15  for i in a:
16      print(i)
17
18  print("\n")
19  for gholi in a:
20      print(gholi)
21
```

```
0
1
2
3
4


0
1
2
3
4


0
1
2
3
4
```

In [5]:
```python
a = "hello"
for i in a:
    print(i)

print("\n")
a = (1, 2, 3)
for i in a:
    print(i)

print("\n")

a = {1, 2, 3, 4}
for i in a:
    print(i)
```

```
h
e
l
l
o


1
2
3


1
2
3
4
```

In [6]:
```python
for i in "hello":
    print(i)

print("\n")
for i in [1, 2, 3]:
    print(i)
```

```
h
e
l
l
o


1
2
3
```

```
In [10]:    1  # Nested for loop
            2  a = [[1, 2, 3], [4, 5, 6]]
            3
            4
            5  for i in a:
            6      print(i)
            7      for j in i:
            8          print(j)
```

```
[1, 2, 3]
1
2
3
[4, 5, 6]
4
5
6
```

```
In [12]:    1  a = [
            2      [[1, 2, 3], [4, 5, 6]],
            3      [[7, 8, 9], [10, 11, 12]],
            4      ]
            5
            6  for i in a:
            7      print(i)
            8      for j in i:
            9          print(j)
           10          for k in j:
           11              print(k)
```

```
[[1, 2, 3], [4, 5, 6]]
[1, 2, 3]
1
2
3
[4, 5, 6]
4
5
6
[[7, 8, 9], [10, 11, 12]]
[7, 8, 9]
7
8
9
[10, 11, 12]
10
11
12
```

In [16]:
```python
dict1 = {'one': 1, 'two': 2, 'three': 3}
# dict1[0]  # False
print(dict1['one'])

# 3 methods for elements in dictionary
# items()
a = dict1.items()
print(a)

print("\n")

# keys()
a = dict1.keys()
print(a)

print("\n")

# values()
a = dict1.values()
print(a)
```

```
1
dict_items([('one', 1), ('two', 2), ('three', 3)])


dict_keys(['one', 'two', 'three'])


dict_values([1, 2, 3])
```

In [20]:
```python
a = ['one', 'two', 'three'] # keys
b = [1, 2, 3]   # Values

print(list(zip(a, b)))
# [('one', 1), ('two', 2), ('three', 3)]

```

```
[('one', 1), ('two', 2), ('three', 3)]
```

In [24]:
```python
dict1 = {'one': 1, 'two': 2, 'three': 3}

# print(dict1['two'])

for i in dict1:
    print(i)
    print(dict1[i])
    print("\n")
```

```
one
1


two
2


three
3
```

In [29]:
```python
dict1 = {'one': 1, 'two': 2, 'three': 3}

for i in dict1.keys():
    print(i)

print("\n")

for i in dict1.values():
    print(i)

print("\n")


# dict_items([('one', 1), ('two', 2), ('three', 3)])
for i, j in dict1.items():
    print(i, ':', j)
```

```
one
two
three


1
2
3


one : 1
two : 2
three : 3
```

In [30]:
```python
1  a = ['one', 'two', 'three'] # keys
2  b = [1, 2, 3]   # Values
3
4  for i, j in zip(a, b):
5      print(i, ':', j)
```

```
one : 1
two : 2
three : 3
```

In [33]:
```python
1  for i, j in ((1, 2), (3, 4)):
2      print(i, j)
3
4  # cause Error
5  a = [[1, 2, 3], [4, 5, 6]]
6  for i, j in a:
7      print(i, j)
```

```
1 2
3 4
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-33-21e2c36ce5a4> in <module>
      4
      5 a = [[1, 2, 3], [4, 5, 6]]
----> 6 for i, j in a:
      7     print(i, j)

ValueError: too many values to unpack (expected 2)
```

In [44]:
```python
1  dict1 = {'one': 1, 'two': 2, 'three': 3}
2  # dict_items([('one', 1), ('two', 2), ('three', 3)])
3
4  for t in dict1.items():
5      i, j = t
6      print(i)
7      print(j)
```

```
one
1
two
2
three
3
```

In [58]:
```python
1  # range(1, 5) 1 2 3 4
2  print(range(1, 5))
3
4  print(list(range(1, 102, 5)))
5
6  print("\n")
7  print(list(range(10, 0, -2)))
```

```
range(1, 5)
[1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 76, 81, 86, 91, 96,
101]


[10, 8, 6, 4, 2]
```

In [63]:
```python
1  for i in range(1, 10):
2      print(i)
3
4  print("\n")
5  a = [1, 2, 3]
6  print(len(a))
```

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


3
```

In [64]:
```python
1  a = [10, 20, 30, 40]
2
3  for i in range(len(a)): # range(0, 4) range(4)
4      print(i)
```

```
0
1
2
3
```

In [76]:

```
1  a = ['one', 'two', 'three'] # keys
2  b = [1, 2, 3]   # Values
3
4  print(len(a) == len(b), len(a))
5
6
7  # dict1 = {'one': 1, 'two': 2, 'three': 3}
8
9  # dict1["one"] = 1
10 # print(dict1)
11
12 dict1 = {}
13 for i in range(len(a)): # 0 1 2
14     dict1[a[i]] = b[i]
15
16 print(dict1)
```

```
True 3
{'one': 1, 'two': 2, 'three': 3}
```

In [85]:
```python
a = [10, 20, 40, 50, 70]
"""
# even index: 0-->10 2-->40 4-->70    a[0] + a[2] + a[4]
# odd index: 1-->20 3-->50            a[1] + a[3]

1 3 5 7 ... 2*n+1
2 4 6 8 ... 2*n
"""


a_even = []
a_odd = []

range_a = list(range(len(a)))
print(range_a)

for i in range(len(a)): # 0 1 2 3 4
    if 2*i in range_a:
        a_even.append(a[2*i])     # a[0] a[2] a[4] a[6] a[8]
    else:
        pass

    if 2*i+1 in range_a:
        a_odd.append(a[2*i+1])    # a[1] a[3] a[5] a[7] a[9]
    else:
        pass

sum_even = sum(a_even)
print(sum_even)
sum_odd = sum(a_odd)
print(sum_odd)
```

```
[0, 1, 2, 3, 4]
120
70
```

while

```python
while True:
    instruction
```

In [ ]:
```python
a = 1
while a == 1:
    print("hi")
```

In [86]:
```python
a = 1
while a <= 10:
    print("hi")
    a += 1
```

```
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
```

In [87]:
```python
a = 1
while a <= 10000:
    print(a)
    a += 1
```

```
1
2
3
4
5
6
7
8
9
10
```

In [89]:
```python
a = 1
while a <= 10000:
    print(a)
    a += 1

    if a == 11:
        break
```

```
1
2
3
4
5
6
7
8
9
10
```

```
In [1]:    1  a = 1
           2  while a <= 20:
           3      print(a)
           4      a += 1
           5      if a == 10:
           6          continue
           7      print("Im here")
```

```
1
Im here
2
Im here
3
Im here
4
Im here
5
Im here
6
Im here
7
Im here
8
Im here
9
10
Im here
11
Im here
12
Im here
13
Im here
14
Im here
15
Im here
16
Im here
17
Im here
18
Im here
19
Im here
20
Im here
```

In [2]:
```python
a = 0
while a <= 20:
    a += 1
    if a == 10:
        continue
    print(a)
```

```
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
19
20
21
```

In [ ]:
```python

```