List

```
names = ['Thomas', 'Steven', 'Jane', 'Tom', 'Susan']
```

0	1	2	3	4
Thomas	Steven	Jane	Tom	Susan

-5	-4	-3	-2	-1
Thomas	Steven	Jane	Tom	Susan

Modifying list

```
# Insert an item at the end of the list
names.append("John")

# Insert an item at specific index
names.insert(0, "Peter")

# Extend list with a list of items.
names.extend(['Victoria', 'Terence', 'Eric', 'Nicolas'])

# delete first occurrence by value.
names.remove('Thomas')

# delete an item by index.
del names[1]
```

List Slicing

```
sample_list[ start : end : step ]

# First three items
sample_list[:3]

# Last 3 items
sample_list[-3:]

# Every second item from beginning to end
sample_list[::2]

# A copy of the whole list
sample_list[:]

# A copy of the whole list, in reversed order
sample_list[::-1]
```

Dictionary

```
sample_dict = {
    'name': 'Thomas',
    'score': 65
}
```

List of dictionary code example

```
students = [
    {'name': 'Thomas', 'score': 65},
    {'name': 'Alan', 'score': 95},
    {'name': 'Jane', 'score': 85},
    {'name': 'Susan', 'score': 75},
    {'name': 'Chris', 'score': 45}
]
```

Nested list and dictionary code example:

```
student_a = {
    'name': 'Steven',
    'email': 'steven@example.com',
    'classes': [
        'Introducing Python',
        'Web scraping with BeautifulSoup',
        'Plotting graph with MatPlotLib'
    ]
}
```

if-condition

```
score = 61

if score >= 60:
    print("Pass")
else:
    print("Fail")
```

Putting logic flow inside function

Function is a block of code that takes **parameters** and **return** calculation result. This block of code is executed only when other line of code **calls** it.

```
def is_student_pass(score):
    if score >= 60:
        return "Pass"
    else:
        return "Fail"
```

Loop and repeat

We can repeat a block of code by using while-loop or for-loop.

While-loop is usually used for iteration that we don't know the total count.

For-loop is when we know the iteration count. For example, we already have the list of items. Or we already know how many times to repeat.

while-loop

```
tasks = []

while True:
    value = input("Please input a to-do task, or 'q' to quit. ")

    if value == 'q':
        break

    tasks.append(value)

print(tasks)
```

for-loop

```
data = ['Apple', 'Banana', 'Orange']

for fruit in data:
    print(fruit)
```

range(start, end, step)

```
# print("0-9")
for i in range(10):
    print(i)

# print("1-10")
for i in range(1,11):
    print(i)

# print("1,3,5,7,9")
for i in range(1,10,2):
    print(i)

# print("2,4,6,8,10")
for i in range(2,11,2):
    print(i)
```