

Lists Store multiple items in a single variable

```
snow = [0.3, 0.0, 0.0, 1.2, 3.9, 2.2, 0.8]

# Index
day_1 = snow[0]      # 0.3
day_7 = snow[6]      # 0.8

# Negative index
day_7 = snow[-1]     # 0.8

# Slicing
snow_weekday = snow[0:5]
snow_weekend = snow[5:7]
```

List Functions & Methods Update lists or analyze list items

```
film_runtimes = [121, 142, 131, 124]

# Built-in functions
len(film_runtimes)    # Output: 4
max(film_runtimes)    # Output: 142
min(film_runtimes)    # Output: 121

# Built-in methods
film_runtimes.append(152)
# [121, 142, 131, 124, 152]
film_runtimes.insert(3, 138)
# [121, 142, 131, 138, 124, 152]
film_runtimes.remove(142)
# [121, 131, 138, 124, 152]
film_runtimes.pop(0)
# [131, 138, 124, 152]
```

Functions Define once to use multiple times

```
def greetings():
    return 'Hello, World!'

print(greetings()) # Output: Hello, World!
```

Parameters & Arguments Affects outcome of functions

```
def add(x, y):
    return x + y

print(add(2, 3))    # Output: 5
print(add(21, 56)) # Output: 77
```

Scope Where in the program a variable is visible

```
t = 29      # Global scope

def func():
    t = 42   # Local scope
    print(t)

print(t)    # Output: 29
func()      # Output: 42
```

Classes & Objects A class is a template for the objects

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def say_hi(self):
        print(f'👋 My name is {self.name}')

brandon = Person('Brandon', 31)
kat = Person('Katherin', 24)

brandon.say_hi() # 👋 My name is Brandon
kat.say_hi()     # 👋 My name is Katherin
```

Modules Python file that can be imported into another

```
from matplotlib import pyplot as plt
import random

print(random.randint(1, 10))

x = [1, 2, 3]
y = [4, 6, 8]

plt.plot(x, y)
plt.show()
```

Notes