Elements of Programming

Gang Chen

chengangcs@gmail.com

Outline

- Data types and Operations
- Flow control
- Functions and Modules
- Managing Python files

Program = Data Structure + Algorithm

Programming = Designing and Implementing of Data Structure and Algorithm

Lecture 5 : Data Structures and Algorithms

1. Basic data types are bricks to build advanced data structures.

2. Flow control statements are the language to describe algorithms.

3. Functions and modules help us organize the source codes of a program.

4. OOP tells us how to design the architecture of your program to integrate data structures and algorithms.

Lecture 3 covers 1-3, and 4 is covered in Lecture 4.

Data Type and Operations

Simple Data Types

- Type
- Value

- Class
- Object

Booleans

Possible Values:

- True
- False

Logical Operations:

- not
- and
- or

Empty Values

Following values are treated as False in logical operations:

- None
- 0
- 0.0
- ""
- ''

Integers

Python 2

- int, limited by the size of a C long (typically 32 or 64 bits)
- long, limited only by available memory

Python 3

- long

Implementation of long in CPython

```
struct _longobject {
    PyObject_VAR_HEAD
    digit ob_digit[1];
};
```

Ref: https://www.wikiwand.com/en/Arbitrary-precision_arithmetic

Operations

- +, -, *, /
- +=, -=, *=, /=
- **, //
- ==, !=

Floats

- 2.5
- 1.222e24
- .0002
- 10002.

Operations

```
1.3 - 0.6 == 0.7
False
```

Strings

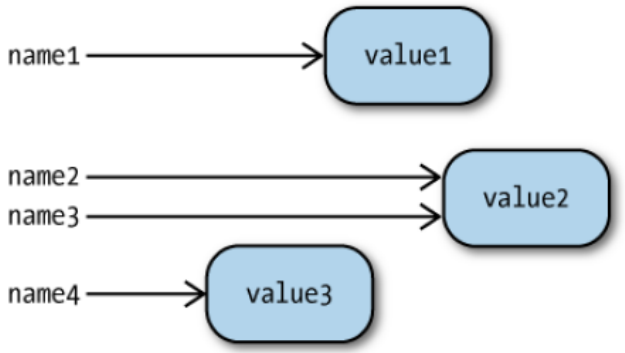
Strings are series of Unicode characters.

- 'Hello World'
- "Hello World"

Operations

- in
- +
- x
- Subscription
- Slicing

Variable



Built-in Data Structures

- sets
- mappings
- sequences
- streams

Flow Control

Conditional Statements

```
if expression:
    statements1
else:
    statements2
```

Conditional Statements

```
if expression1:
    statements1
elif expression2:
    statements2
    # . . . any number of additional elif clauses
else:
    statements
```

Loop

```
while expression:
    statements1
else:xs
    statements2
```

Iterations

```
for item in collection:
    do something with item
```

Exception Handling

```
try:
    try-statements
except ErrorClass:
    except-statements
```

Functions and modules

Functions

Defining Functions

```
def name(parameter-list):
    body
```

Comments and Documentation

ref: https://www.python.org/dev/peps/pep-0257/

Modules

import

```
import name
```

namespace

```
from modulename import name1, name2, ...
from modulename import actualname as yourname
from modulename import *
```

Managing Python Files

- create subdirectories for different modules of your program
- add **init**.py to these directories
- Content of **init**.py

```
# in your __init__.py
from file import File
```