Basic Python Programming

Yilin Zheng

Dept. of Computer Science and Engineering

June 26, 2017

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Outline

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Homebrew

- Homebrew is a package manager for MacOS.
- Please visit ▸ Homebrew
- I wish you can use little Terminal to do something since you will touch it sooner or later.
- Open terminal
- Enter the command line on the website

Figure: terminal

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Install Python

- Enter this command to check the version of installed python

```
python3 -V
```

- If no python installed, enter this line in terminal to install

```
brew install python3
```

- after installed, enter

```
python3
```

then you will get this interface

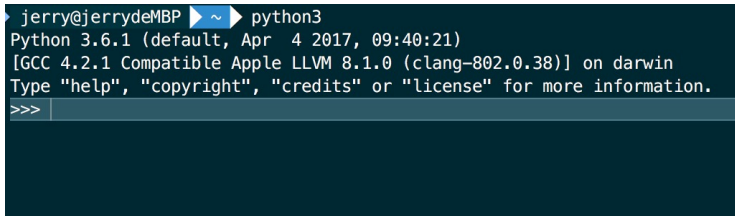

Figure: terminal

Sublime Text

- Also, use text editor like Sublime Text is another way.
- New a file with ".py" as filename suffix(e.g. helloworld.py), then write as following:

```
#!/usr/bin/python
# Filename : helloworld.py
print ('Hello World')
```

- Execute the command

```
python3 helloworld.py
```

to see the result

Constant

- Constant **cannot be changed**
- Represent its literal meaning
- e.g. Enter '2' in Python interface:

```
>>> 2
2
>>> 'Attest'
'Attest'
>>> 100/2
50.0
>>>
```

Numeber

There are fours types of number in python

- int
- long int
- float
- complex

```
>>> type(2)
<class 'int'>
>>> type(2.5)
<class 'float'>
>>> type(2+3j)
<class 'complex'>
```

- Use `type` to get the type.
- For complex, use `j` as Imaginary Unit

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

String

- String is an array of characters, also **cannot be chanegd**
- Use ' ' or " " to indicate single line string

```
>>> 'detest'
'detest'
>>> "detest"
'detest'
>>>
```

- Use ''' ''' or """ """ to indicate multiple lines string

```
>>> '''hello
... hi'''
'hello\nhi'
>>>
```

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Escape character

- Use \ to print out some character

```
>>> 'What\' your name?'
"What' your name?"
>>>
```

- If no \

```
>>> 'What' your name?'
File "<stdin>", line 1
  'What' your name?'
           ^
SyntaxError: invalid syntax
>>>
```

Identifier

Some rules to name an identifier:

- First character should be **letters** or _
- Remaining part can be letters, numbers(0-9) or '_'
- Identifier is case sensitive, it means 'a' is distinguished from 'A'
- e.g.

```
>>> a = 10
>>> 2a =10
File "<stdin>", line 1
2a =10
  ^
SyntaxError: invalid syntax
>>> a
10
>>>
```

Variable

- Variables store value and **can be modified**
- e.g.

```
>>> Name = 'Jerry'
>>> Name
'Jerry'
>>> Name = 'Tom'
>>> Name
'Tom'
>>>
```

- Identifiers of variables should follow the rules mentioned before
- Creat variable directly use unique identifier, no need to use type name, which much different from other languages like C or Java

Data type

- Variables can handle various types of data.
- As we go further, we will be able to create our own data types by using **class**
- Don't take too much about some details we haven't discussed so far

Object

That's why we usually joke about programmers who can create their objects

- Python can solve problem in **Object Oriented** way, which is abbreviated as OOP
- Object describes **attributes of things**, so does Java
- To distinguish from **Function**, function describes the behavior of a process(Process-oriented programming)
- Today, OOP is much popular than POP
- More detail will be present in later slides

Logical line and Physical line

- Physical line **can be seen when programming**
- Logical line is something like `print 'Hello world'` which actually takes up a physical line
- By default, Python wish only one **expression**(disscussed next section) per line, which means a logical line per physical line
- Sometimes, you are willing to put more than one expression in a line, then use `;` to separate them apart

Indentation

- Indentation is **very IMPORTANT** in python which may affect the result
- Four **Space** or one **Tab** as the unit of indentation
- It will go wrong if no indentation before `print`

```
#!/usr/bin/python3
# Filename : LargerThan20.py
a = 10
def SayHello(a):
    if a < 20:
        print ('a is smaller than 20')
    return
SayHello(a)
```

- Codes in the same level have same indentation, we call these codes a **block**

Comment

- Use # to comment on one line
- Use ''' ''' or """ """ for multiple lines comment

```
#!/usr/bin/python3
# Filename : LargerThan20.py
a = 10 # an variable nameed 'a', stores a int
                                value 10
def SayHello(a):
    if a < 20:
        print ('a is smaller than 20')
    return
'''
If a is smaller than 20, print the sentence
'''
SayHello(a) #call the funtion
```

Operator

- Arithmetic operators
  `+` `-` `*` `**`(power) `/` `//`(divisible) `%`
- Comparison(Relational) operators < > <= >= == !=(inequal)
- Bitwise operators
  `<<`(left) `>>`(right) `&`(and) `|`(or) `^`(xor) `~`(not)
- Logical(Boolean) operators `and` `or` `not`
- Assignment operators `=`

```
>>> 2**3
8
>>> 2!=3
True
>>> 2==3 and 2!=3
False
```

Operator priority

- Basic rule in arithmetics
- Logical $<$ Comparison $<$ Bitwise $<$ Arithmetic
- More details plz visit ▸ Operator priority
- Use **parenthesis** ( ) to indicate the order explicitly

Expression

- An expression is an executable codes

```
>>> 5//2
2
>>> 3&1
1
>>> print('This is the end of this slide.')
This is the end of this slide.
>>>
>>>quit() # use this command to exit python
                                    interface
```