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

Constant

- Constant **cannot be changed**

Numeber

Fours types of number

- **int, long int, float, complex**
- Use `type` to return the type
- Use `j` as Imaginary Unit, e.g. `1+2j`

String

- String **cannot be chanegd**
- Use ' ' or " " to indicate single line string
- Use ''' ''' or """ """ to indicate multiple lines string, also can be used to comment

Escape character

- Use \ to print out some character

Escape	What it does.	含义
\\	Backslash ()	反斜杠
\'	Single-quote (')	单引号
\"	Double-quote (")	双引号
\a	ASCII bell (BEL)	响铃符
\b	ASCII backspace (BS)	退格符
\f	ASCII formfeed (FF)	进纸符
\n	ASCII linefeed (LF)	换行符
\N{name}	Character named name in the Unicode database (Unicode only)	Unicode数据库中的字符名；name就是它的名字
\r ASCII	Carriage Return (CR)	回车符
\t ASCII	Horizontal Tab (TAB)	水平制表符
\uxxxx	Character with 16-bit hex value xxxx (Unicode only)	值为16位十六进制xxxx的字符
\Uxxxxxxxx	Character with 32-bit hex value xxxxxxxx (Unicode only)	值为32位十六进制xxxx的字符
\v	ASCII vertical tab (VT)	垂直制表符
\ooo	Character with octal value ooo	值为八进制ooo的字符
\xhh	Character with hex value hh	值为十六进制数hh的字符

Figure: Escape Characters

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Identifier

Some rules to name an identifier:

- First character: **letters** or _
- Remaining part: letters, numbers(0-9) or '_'
- Identifier is **case sensitive**

Variable

- Variables **can be modified**
- Use unique identifier, following the naming rule

Indentation

- Four **Space** or one **Tab** as the unit of indentation
- Codes in **the same level** have **same indentation**, called **block**

Comment

- One line comment: #
- Multiple lines comment: ''' ''' or """ """

Operator

- `**` (power) `//` (divisible)
- == !=(inequal)
- `<` `<` (left shift) `>` `>` (right shift) `&` (and) `|` (or) `^` (xor) `~` (not)
- `and` `or` `not`

Operator priority

- Logical < Comparison < Bitwise < Arithmetic
- More details in ▸ Operator priority
- Use **parenthesis** ( ) to indicate the order explicitly!!!

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Conditional statement

- Conditional statement has **Boolean value**
- Boolean value: `True` `False`

```
>>> 100//4 == 5**2
True
>>> not True
False
>>> 2+3 > 10
False
>>>
```

- Use conditional statements to decide whether branch or loop will take place

if and else

- Use `if` to create conditional branch, takes an **conditional statement**, ends with an **colon** `:`
- Use `else` to give another choice
- Example, execute the block if conditional statment is `True`

```
#!/usr/bin/python
#Filename: if.py
text1 = 'The night is young\n...' #\n skip to
                                  next line
Song_Name = 'Here With You'
if(Song_Name == 'Here With You'):
#when the value of the condition is True, go
                                  into its block
    print (text1)
else:  #if False, skip to this block
    print ('Aha, no matched lyric.')
```

elif

- `elif` means **else if**

```
#!/usr/bin/python
# Filename: if.py
a = 25
b = 25
if a == b:
    print('a == b')
elif a < b:
    print('a < b')
else:
    print('a > b')
```

while

- Use `while` to implement **Loop**, takes an **conditional statement**, also ends with an colon
- Loop will be exectuted if `True`, else goes to next block

while

```
#!/usr/bin/python # Filename: while.py
number = 26
running = True
while running:
    guess = int(input('Enter an integer : '))
    if guess == number:
        print('Congratulations, you guessed it.')
        running = False # this causes the while
                        loop to stop
    elif guess < number:
        print('No, it is a little higher than that
                        ')
    else:
        print('No, it is a little lower than that'
                        )
print('The while loop is over.')
```

for

- `for` is another key word for loop which usually need to **indicate a range**
- Use `in...range()` to indicate the range
- e.g. `for i in range(1,5)`
  This code will loop from 1,2,3,4 but not 5
- `range()` takes two number as one is the beginning, the second is the end(excluded)
- By defalut, the step length is **1**, but with `range()` taking the third number, we can modified the step length. e.g.
  `for in range(1,5,3)` will loop from 1 and 4

for

```
#!/usr/bin/python
# Filename: for.py
for i in range(1, 5):
    print(i**2)
print('Done.')
```

```
#!/usr/bin/python
# Filename: for.py
for i in range(1, 100, 4):
    print(i)
print('The for loop is over.')
```

break

- `break` is used to **stop** and **escape from loop**
- Can be used both in `while` and `for` loop

```
#!/usr/bin/python
# Filename: break.py
while True:
    s = input('Enter something : ')
    if s == 'q':
        break
    print('Length of the string is', len(s))
print('Over.')
```

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

continue

- This key word `continue` is to **skip the subsequent codes in current loop and directly execute next loop**
- Example is much vivider to explain its use

```
#!/usr/bin/python
# Filename: continue.py
while True:
    s = input('Enter something : ')
    if s == 'q':
        break
    if len(s) < 3:
        continue
    print('Sufficient input length is', len(s)
                                    )
print('End of both the loop and this slides')
```