程式設計概論
Programming 101

—程式變數與第一個python內建函式 (Variables)

授課老師：邱淑怡

DATE: 9/20/2023

Outline

1. Python coding style
2. Data type
3. Variable
4. How to use variables

Python coding style

- Program: contains lines of statements
- Statements: keyword, special character, identifier
  - Keyword: words with specific meanings or usages defined by Python. ex: if, for, while, …
  - Special character: (), [], {}, “’, #,
  - Identifier(識別碼): new words named by a programmer for defining a new variable or function.
- **Uppercase and lowercase characters are different in Python.**
- Avoid meaningless spaces
- Use meaningful words as variable names

Python coding style (cont.)

- Indentation (縮排)
  - Follow indentation rules.
  - Indent a line by pressing the Tab key at the beginning of the line
- Comment (註解)
  - #: one line comment
  - ”’ or ”””: Multi-line comment
- It is recommended to describe a statement per line.
  - print(‘Python’); print(“Python code”)

Python data type

- Numeric type
  - int, float, bool(boolean)
- Text sequence (or string) type
  - string
- sequence type
  - list, tuple
- Non-sequence type
  - set type: set
  - mapping type: dictionary (dict)

data type

- Numeric type
  - int: integer
    - print(3+6)
  - float: floating point
    - Ex: 3.14159265359
  - bool: Boolean: True (T) / False (F)
    - 1>2
- str: string. consists of characters or texts.
  - single quotes (‘), double quotes(“), three single quotes(‘’’), three double quotes(“””)
  - **Use single quotes and double quotes separately**

different data type
different ways to use them

Pay attention to capitalization

Variable naming

The first character of the variable cannot be any number

- Every variable has its name that represents a value stored in the computer's memory.
- Rules of naming variables
  1. The first character should be the English alphabet or underscore(_), then, there is the English alphabet, numbers(0,1…9), or underscore after the first character.
  2. cannot use Python keywords and built-in functions as variable names.
  3. **Uppercase and lowercase characters are different.**
  4. Variable names do not use Chinese words.
  5. Suggest using meaningful words as variable names.

Variables

- Variables are used to store numbers, strings, or other data types.
- Use assignment(=) to create a new variable

a=10

b=3.56

c="Python for everyone"

message = '近來可好'

Python special usage:
chained assignments
X=Y=Z=100
a, b, c=100, 3.14159, "Hello"

「 = 」:assign

- Variable names are on the left side of the equal sign
- The confirmed data(number or string) are on the right side of the equal sign

Python variable incorrect example

7eleven=‘7-11’

Q&A=‘Q&A’

Q A =10

if=‘if’

Python functions

- Function: Provides many useful tools
  - The build-in functions or defined functions in the data type class
  - The module functions
  - The defined functions by myself
- Example: print() function

print(item1[,item2,…, sep=separate characters , end=end characters])

* default value: sep=‘ ‘,end=‘/n’, []:Options

print(100,”Python”,60,sep=‘###’,end=‘.’)

print() function for format output

1. format by % method
2. format by format method
3. f-string method

print() function using % method

print(Strings % (parameter list))

* %s: string, %d: int, %f: float, aligning right by default
* %3d :3 places for int, %3s: 3 spaces for strings
* %-3d or %-3s: 3 places for int or string with aligning left
* %.2f: float with 2 decimal places
* %6.2f: 6 spaces (3 spaces with int, 1 decimal point, 2 digits come after the decimal point)

print("請印出年齡:%d歲，修課名稱:%s"%(20,'python'))

parameter	meaning
%d	int (integer)
%s	string or text
%f	float
%%	string with %

print() function using format method

print(string.format(parameter list))

* {}: Parameter(參數) order
* {i}: i is the order of parameters, for example: {0} is the first one
* {i:format}：the format is like to %

```
print("{1}考試成績是{0:8.2f}分".format(78.9086,"John"))
```

print() function using f-string method

```
text = ‘python’
print(f'Hello, {text}’)


x = 10
y = 27
 print(f'x + y = {x + y}')
```

Review

Textbook: chapter 2: 2.2, 2.3, 2.4, 2.5, 2.8