Lesson 3 Python Basic Syntax

This lesson will explain the basic syntax of Python, such as comments, indentation rules, coding standards, etc.

1. Comment

Comments are used to explain Python code. Python support two types of comments, including single line comment and multi line comments.

1) Single Line Comment

Comment starts with “#” and its format is as follow.

```
# comment
```

2) Multi Line Comments

Insert three single quotation marks “'''” or three double quotation marks “"""” at the beginning and the end of the comment to comment multiple lines, and the format is as follow.

```
'''
comment
comment
'''
```

```
"""
comment
comment
"""
```

2. Indentation Rules

Python uses indentation and colon symbol (:) for showing where blocks of code begin and end

In Python, for class definitions, function definitions, flow control statements, exception handling statements, etc., the colon at the end of the line and the indentation of the next line represent the beginning of the next

code block. And when indentation ends, a block of code ends.

Each red frame in the picture below represents one block of code.

```
1  if num>100
2      print("num>100")
3  if num>50 and num<=100:
4      print("50<num<=100")
5  if num<=50:
6      print("num<=50")
```

Note: block of code at the same level should be indented consistently. We can use "**Tab**" key or input 4 spaces to indent.

Note: Python uses 4 spaces as indentation by default. In general, one "Tab" is equal to 4 spaces.

---

3. Coding Standard

Python adopts PEP8 as coding standard. "**PEP**" represents Python Enhancement Proposal, and "**8**" indicates style guide of Python code.

Please strictly follow the coding standard when coding to make the code neater, which will enhance the readability.

1) One "import" is for one module. Please don't import multiple modules for one time.

Recommend	`import os` `import sys`
Not Recommend	`import os,sys`

2) Please don't put semicolon ";" at the end of the line, and don't put two commands at the same line

Recommend	print("Hello World") print("Hello Hiwonder")
Not Recommend	print("Hello World");print("Hello Hiwonder")

3） The length of line should not be greater than 80 characters, and you can separate a command into several lines, and put the command inside "**()**", as the example shown below. It is not recommended to use backslash "\" to connect the lines of contents.

Recommend	a=("Python is a multi-platform programming language," "provide ample API and tools")
Not Recommend	a="Python is a multi-platform programming language, \ provide ample API and tools"

4） When necessary, we can input space to improve the readability of the code.

5） In general, it is recommended to input space around both sides of operators and commas, and between parameters of function to separate.

4. Identifier Naming Standard

Identifier is the name of variable, function, class, module and other objects. In Python, naming identifier should be consistent with the naming rules.

1） Identifier names in Python can contain letters (A~Z, a-z), underscore (_) and number, and the name should always start with a non-numeric

character.

2）Identifier must be different from keywords/ reserved words in Python. For the definition of keywords/ reserved, please move to "**5. keyword/ reserved word**".

3）Identifier cannot contain **space**, **@, % , $** and other special characters.

结合上述三点规则，下表列举了部分命名合法的标识符与不合法的标识符：
Examples of valid and invalid identifier are listed below.

Valid Identifier	Hiwonder mode01 user_age

Invalid Identifier	`4world # Number cannot be the first character`
	`try # try is the reserve word and can not used as identifier`
	`$money # special character cannot contained`

4）Identifier is case sensitive. For example, "num", "Num" and "NUM" are three independent variables.

5）Identifier starting with underscore "_" has special meaning. Please avoid using identifier starting with "_" if not necessary.

Identifier	Meaning	Example
Start with single underscore	Class properties that cannot be accessed directly through "from...import*"	_width
Start with double underscore	Exclusive member of class	__add

Start and end with double underscore	Special identifier	__init__

6）Chinese can be used as identifier in Python. To avoid error, please shun Chinese.

Besides the rules mentioned above, there are corresponding rules in identifier naming under different situation.

1）When used as module name, identifier should be short and composed of lower case letters. And underscore "_" can be used for separation.

2）When used as package name, identifier should be short and composed of lower case letters, but it is not recommended to use full stop ".", such as "com.mr" and "com.mr.book".

3）When used as class name, identifier should start with upper case letters, for example "Book" which defines a book class.

4）When used as class name inside module, identifier can start with "_" and upper case letters, for example "_Book".

5）When used as function name, property name and method name in classes, identifier should be composed of lower case letters and underscore can be used to separate different words.

6）When used as constant name, identifier should consist of upper case letters and different words can be separated by underscore "_".

5. Keyword/ Reserved Word

Reserved word, also called keyword, is a word with special meaning in

Python. And they cannot be used as variable names, function names, class names, module names or any other object names

In Python interactive programming environment, we can check the reserved words according to the steps below.

1） Start virtual machine, and click  , and then click  or press "**Ctrl+Alt+T**" to open command line terminal.

2） Input "**python3**" command and press Enter to enter Python interactive programming environment.

```
hiwonder@hiwonder-virtual-machine:~$ python3
```

3） Input "**import keyword**" command and press Enter to import "**keyword**" module.

```
>>> import keyword
```

4） Input "**keyword.kwlist**" commend and press Enter to view all reserved words in Python.

```
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del',
'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',
 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Note: Reserved words are also case sensitive. For example, "if" is reserved word, while "IF", "iF" and "If" are not.

---

6. Data Type

There are six types of data in Python3, including Number, String, List, Tuple, Dictionary and Set.

And Number, String and Tuple are immutable, while List, Dictionary and

Set are mutable.

Note: In Python, type() function can be used to check the type of variable.

6.1 Number

Number includes three numeric types to represent numbers or value.

1） int: Integers can be binary, octal, decimal and hexadecimal values.

2） float: floats are decimal values generally composed of integers and decimals. Each floating point number occupy 8 bytes, that is 64 bits.

3） complex: complex number is a number with real and imaginary components. Both real and imaginary components belong to floating type.

4） bool: Only has “**True**” and “**False**” values. “**True**” corresponds to “**1**” and “False” corresponds to “**0**”

6.2 String

A string is a collection of multiple characters. Strings in Python are surrounded by either single quotation marks “**'**”, or double quotation marks “**"**” and triple quotation marks “**'''**” or “**"""**”.

Strings with single quotation marks and double quotation marks are equivalent, and return the objects of the same type.

Note: when there is quotation marks inside the strings, we need to escape them through adding backslash “\” in front of the quotation marks to avoid syntax error.

6.3 List

List is a sequence structure in Python, which can store any types of data,

including integer, decimal, string, list, tuple, etc. Its format is as follow.

```
Variable name = [element 1, element 2,..., element n]
```

The number of elements in List is unlimited and there can be different types of elements in one List. For better readability of program, it is recommended to use one type of data in one list.

Each element in the List corresponds to integer index value. The corresponding element values can be obtained though index values so as to change and delete them.

6.4 Tuple

Tuple is another important sequence structure in Python, which is similar to List. And its format is as follow.

```
Variable name = (element 1, element 2,..., element n)
```

Different from List, Tuple is immutable sequence, which means that its elements cannot be changed or deleted.

6.5 Dictionary

A dictionary is an unordered, mutable sequence that is created in the following format.

```
Variable name = (key1:value1, key2:value2,…, keyn:valuen)
```

Dictionary is the only type of mapping in Python, which means that elements corresponds to each other.

The elements of dictionary can be List, Tuple, Dictionary and other types of data, but key value must be the immutable type. In addition, there should be only one key value in the same dictionary variable.

6.6 Set

Set is used to store non-repetitive elements and its format is as follow.

```
variable name = {element 1, element 2,..., element n}
```

Set can only store immutable data type, including integer, float type, string and tuple, but cannot store mutable data type, including List, dictionary and set.