第一讲
准备工作
Lecture 1
Getting Started

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

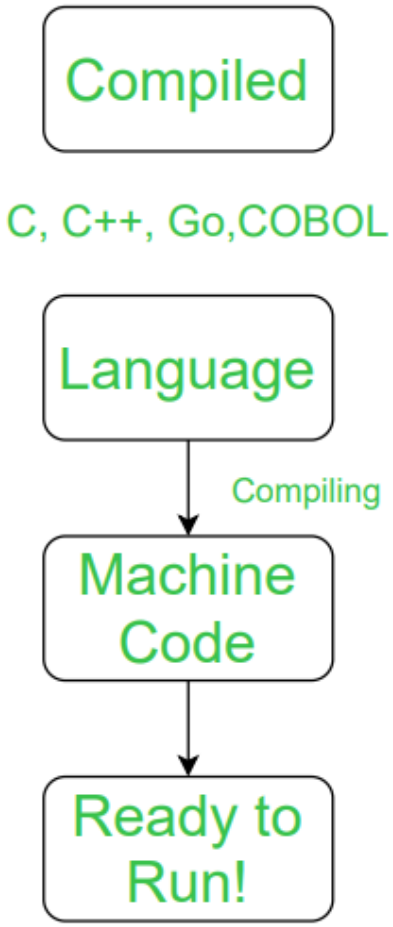
语言分类
Categories of Programming Languages

- 计算机程序设计语言主要分为编译型语言与解释型语言。

  Computer programming languages are mainly categorized into compiled languages and interpreted languages.

- 编译语言需要经由编译，链接，将代码最终由目标机器指令表达的程序设计语言。编译链接后，原则上语言与编程环境不再有关系，即编译好的代码可直接运行。编译语言的种类有C、C++、Fortran、COBOL 等。

  The code of the compiled language needs to be compiled and linked, and finally expressed in the form of target machine instructions. In principle, the language is independent of the programming environment after compiling and linking, that is, the compiled code can be run directly. Compiled languages include C, C++, Fortran, COBOL, etc.



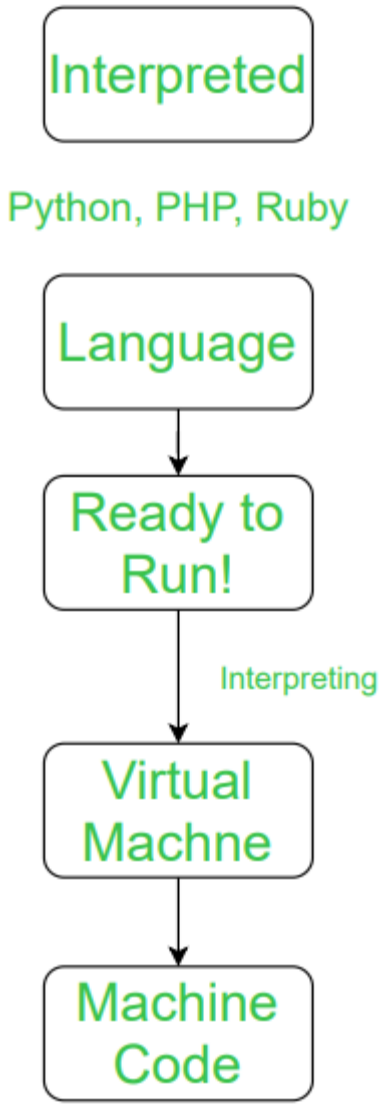
语言分类
Categories of Programming Languages

- 解释型语言通常不需要将程序编译成机器指令，而是由其他程序—如解释器—读取和执行；解释型语言书写的程序通常不能独立地运行。解释语言包括 JavaScript、Perl、Python、BASIC 等。

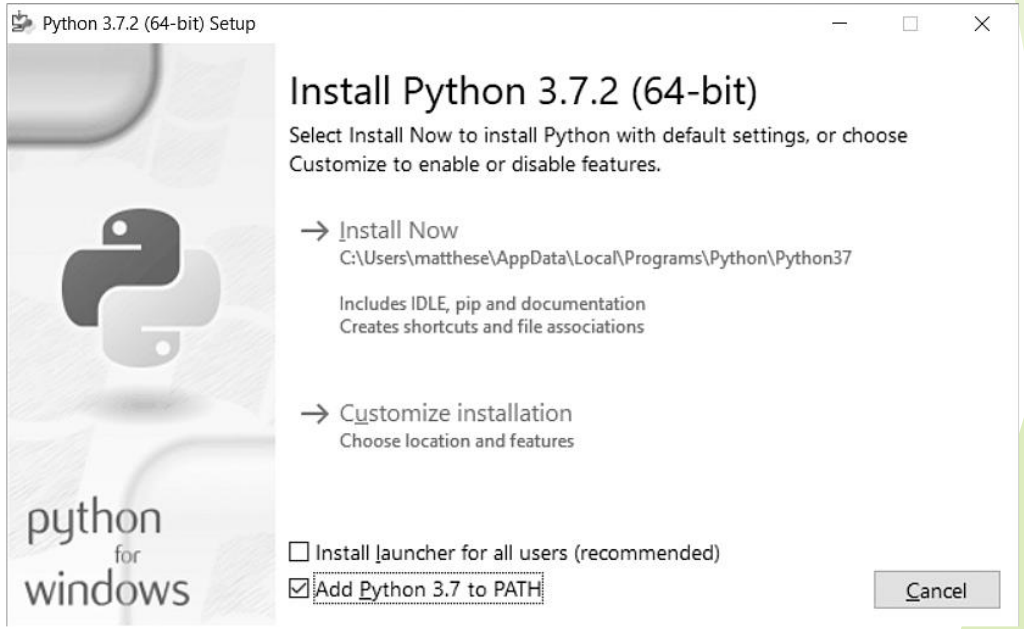
  Interpreted languages usually do not need to compile programs into machine instructions, but are read and executed by other programs, such as an interpreter; programs written in interpreted languages usually cannot run independently. Interpreted languages include JavaScript, Perl, Python, BASIC, etc.

- Python属于解释型语言。

  Python belongs to the category of interpreted language.



Windows下Python编程环境
Python on Windows



- 基于官方发行包与cmd窗口的使用：

  Start from the official release package and command window

  - 从官网下载安装包：

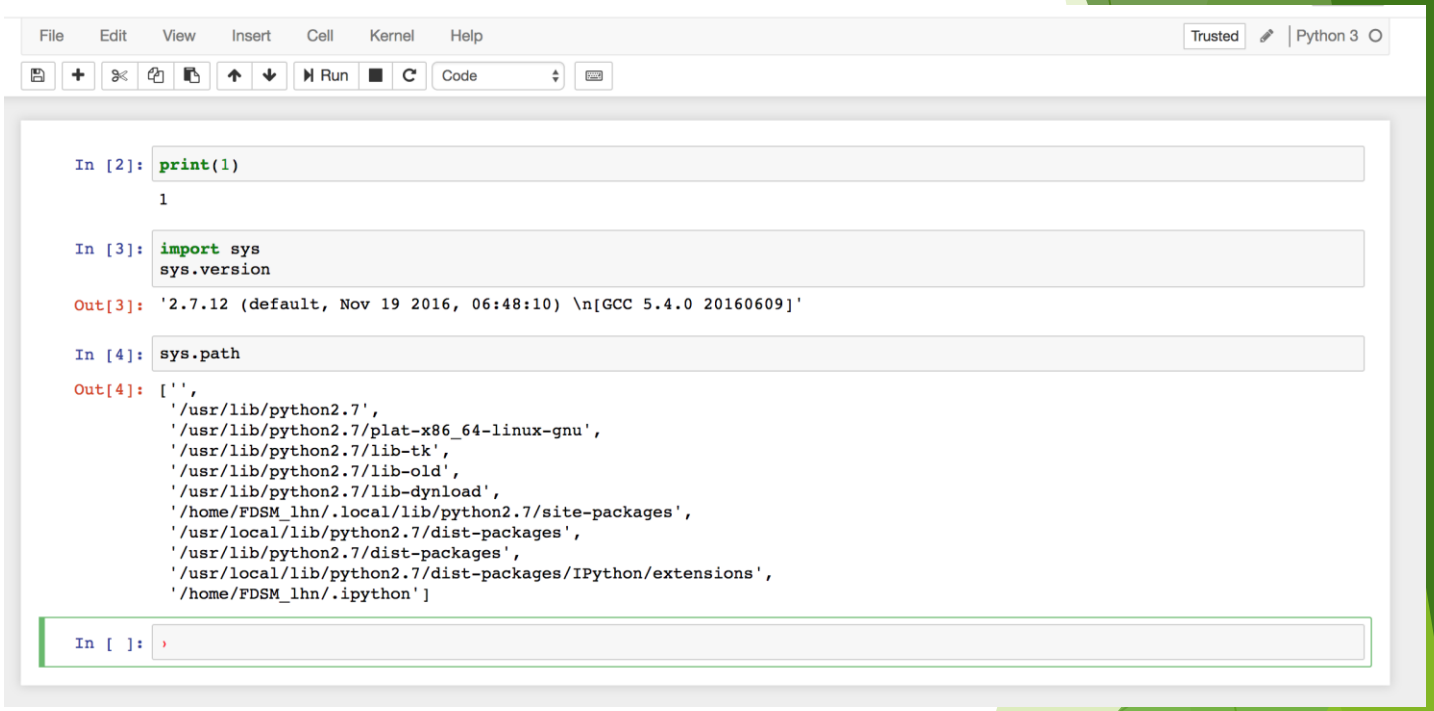
    Download the installation package from the website:

    https://www.python.org/downloads/

  - 安装、配置、使用，必要时可创建虚拟环境。

    Install, configure and use. Create a virtual environment if necessary.

Windows下Python编程环境
Python on Windows



- 基于Intel发行包与浏览器的使用：

  Start from the Intel Python distribution and web browser

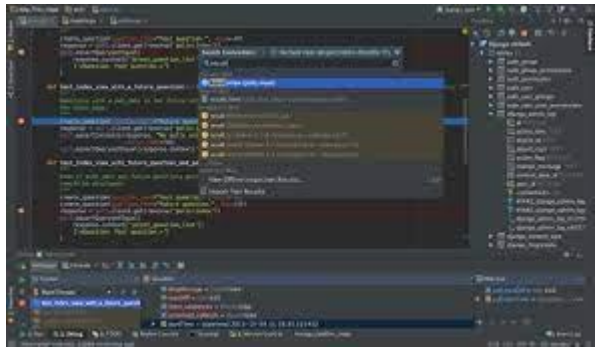
  - 从官网下载安装包：

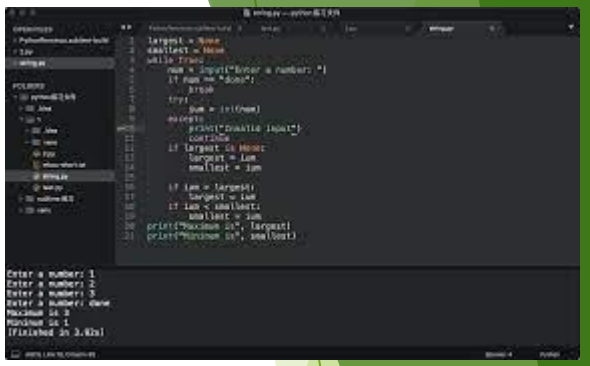
    Download the installation package from the website:

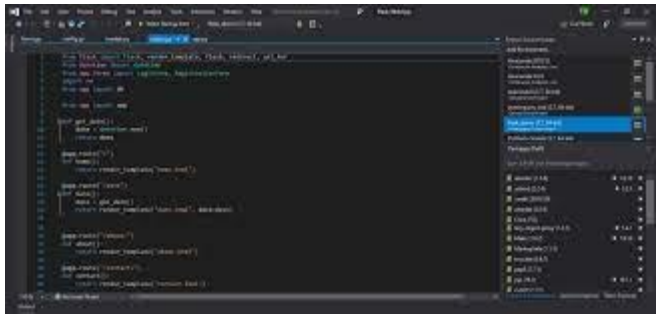
    https://www.intel.com/content/www/us/en/developer/tools/oneapi/distribution-for-python.html

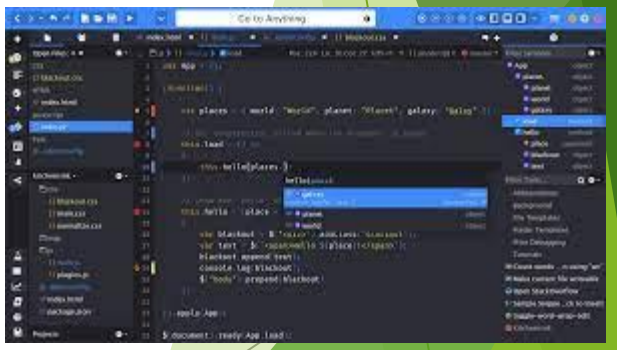
  - 安装、配置、使用，必要时可创建虚拟环境。

    Install, configure and use. Create a virtual environment if necessary.









Windows下Python编程环境
Python on Windows

- 其它可选方式（Other options）：
  - PyCharm：

    https://www.jetbrains.com/pycharm/
  - Sublime：

    https://www.sublimetext.com/
  - Visual Studio：

    https://visualstudio.microsoft.com/
  - ActivePython：

    https://www.activestate.com/products/python/

运行Python
Running Python in a Terminal Session

- 当Python安装与配置好之后，有两种方式可以使用Python，即启动Python解释器，以交互方式运行Python代码，或者直接执行由Python代码组成的文件。如果是集成开发环境，则通过菜单选项一般可启动脚本的执行。而通过使用jupyter notebook的方式，其执行介于交互式执行与直接运行之间。

  After Python is installed and configured, there are briefly two ways to use it. The first is to run the Python code interactively, after launching the Python interpreter. The second is to directly execute the file composed of the Python code in the way of command line. The execution of the script in an integrated development environment (IDE) can generally be started through a menu entry or tool button, such as “run”. By using jupyter notebook, its execution is some way between interactive execution and direct execution.

```
>>> print("Hello Python interpreter!")
Hello Python interpreter!
>>>
```

```
C:\Desktop\python_work> python hello_world.py
Hello Python world!
```

语言分类
Categorization of Programming Languages

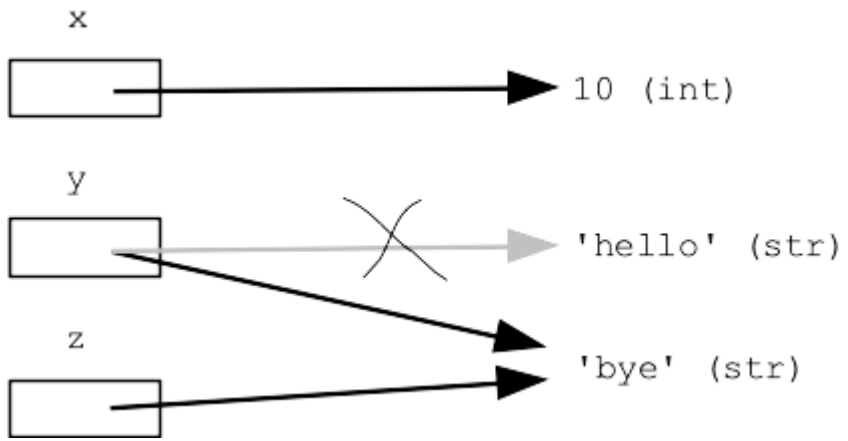
- 在计算机编程中，对编程语言进行分类的另一种方式，是依据语言的类型系统，分成强类型语言和弱类型（松散类型）语言。不过，对于这些术语的定义并不存在精确的技术定义，并且不同的作者对术语的隐含含义，以及主流编程语言类型系统的“强度”并没有一致的认识。
- 一般来说，强类型语言，如果同时又是静态类型，则强调在编译时有更严格的类型规则，即在编译期间可以发现与类型规则相左的错误和异常。如果是动态类型，则强调在运行时有更严格的类型规则。大多数规则影响变量赋值、函数返回值、过程参数和函数调用等。
- 弱类型语言，一般都是动态类型语言，类型具有更宽松的类型规则，常常在运行时执行隐式类型转换，可能会产生不可预测甚至错误的结果。

语言分类
Categorization of Programming Languages

- In computer programming, one of the many ways that programming languages are colloquially classified is whether the language's type system makes it strongly typed or weakly typed (loosely typed). However, there is no precise technical definition of what the terms mean and different authors disagree about the implied meaning of the terms and the relative rankings of the "strength" of the type systems of mainstream programming languages.
- Generally, a strongly typed language, if it simultaneously static typed, has stricter typing rules at compile time, which implies that errors and exceptions are more likely to happen during compilation. In contrast, a strongly but dynamic typed language means higher occurrence of errors and exceptions during runtime. Most of these rules affect variable assignment, function return values, procedure arguments and function calling.
- A weakly typed language (usually it is simultaneously dynamic typed) has looser typing rules and may produce unpredictable or even erroneous results or may perform implicit type conversion at runtime.

变量
Variables

- 通俗地讲，我们可以将编程语言中的变量看作一块命名的计算机内存，其中包含着程序运行时的一些信息。打个比方，可以把变量想象成一个有名字的盒子，我们可以在其中“存储”一些东西。根据程序运行需求，代码中可能需要创建、更行和删除变量。变量通常和类型相关联，例如，创建了一个名为 A 且类型为整数的变量。此时，A蕴含了特定的逻辑表示和大小。
- 虽然在最初使用变量的时候，变量可被描述为在其中存储值的盒子。但这种看法不是描述变量在 Python 内部如何表示的准确方法。从另一个角度，将变量视为可以关联值的标签要好得多，或者，也可以说变量引用了某个值。



变量
Variables

- In a plain language, a variable in programming language is a named piece of computer memory, containing some information inside. For example, thinking of a variable as a box with a name, where we can “store” something. We can create, update and delete variables, as much as we need through the source code in accordance with our tasks. Variables usually are of certain types, say, we created a variable with name A and of type integer, which means A defines a certain logical representation and size.
- Although variables are often described as boxes in which values can be stored, this idea can be helpful the first few times when one uses a variable, but it isn’t an accurate way to describe how variables are represented internally in Python. It’s much better to think of variables as labels that values can be assigned to, alternatively, we can also say that a variable references a certain value.

变量命名
Naming Variables

- 变量名只能包含字母、数字和下划线，且以字母或下划线开头，但不能以数字开头。例如，变量 message_1 合法但1_message不合法。

  Variable names can contain only letters, numbers, and underscores. They can start with a letter or an underscore, but not with a number. For instance, message_1 is a valid variable name but not 1_message.

- 变量名中不允许有空格，必要时可以使用下划线分隔变量名中的单词。例如，greeting_message 有效，但是greeting message 会导致错误。

  Spaces are not allowed in variable names; underscores can be used to separate words in variable names if needed. For example, greeting_message works, but greeting message will cause errors.

- 变量名不能使用 Python 关键字和内置函数，避免使用 Python 为特定程序目的而保留的词，例如词 print。

变量
Variables

Avoid using Python keywords and built-in function names as variable names; Try not using words that Python has reserved for a particular programmatic purpose, such as the word print.

- 变量名称应该尽量简短但具有描述性。使用小写字母 l 和大写字母 O 时要小心，因为它们可能与数字 1 和 0 混淆。

  Variable names should be short but descriptive. Meantime, be careful when using the lowercase letter l and the uppercase letter O because they could be confused with the numbers 1 and 0.

- Python变量尽量使用小写。虽然使用大写字母不会出错，但变量名中的大写字母在某些情况下具有特殊含义。

  The Python variables should be lowercase, although it might be okay when using uppercase letters, but uppercase letters in variable names have special meanings in some context.

字符串
Strings

- 在Python中，字符串是一系列字符组成的序列。一般用成对的单引号（或撇号）或双引号框范的字符序列表示。注意这种方式允许在文件中同时使用引号和撇号：

  A string is a series of characters. Anything inside quotes is considered a string in Python, and you can use single or double quotes around your strings. This flexibility allows you to use quotes and apostrophes within your strings:

  "This is a string.“

  'This is also a string.’

  'I told my friend, "Python is my favorite language!“’

  "The language 'Python' is named after Monty Python, not the snake.“

  "One of Python's strengths is its diverse and supportive community."

字符串
Strings

```
>>> s = "PyTHON"
>>> s.upper()
'PYTHON'
>>> s.lower()
'python'
>>> s.title()
'Python'
>>>
```

- Python提供了对字符串操作的一系列函数，如大小写转换，特定格式转换：

  Python provides a series of functions for string manipulation, such as case conversion, specific format conversion.

- 除了普通字符串之外，Python中的串还有所谓的格式化串。格式化串有两类，一类是控制变量替换，二类是控制打印输出。

- In addition to ordinary strings, strings in Python also have format strings or f-strings. There are two types of format strings, one is to control variable replacement, and the other is to control print output.

```
>>> first_name = "Yurui"
>>> surname = "MING"
>>> full_name = f"{first_name} {surname}"
>>> full_name
'Yurui MING'
>>> full_name = "Yurui\nMING"
>>> full_name
'Yurui\nMING'
>>> print(full_name)
Yurui
MING
```

数字
Numbers

```
>>> 0b11
3
>>> 0o71
57
>>> 0xa2
162
>>> +7
7
>>> -1
-1
>>> 1_2345_6789
123456789
>>> 3e11
300000000000.0
>>> 4e-8
4e-08
>>> 5e-2
0.05
```

- Python中的数字包括整数与浮点数。对于整数有二进制，八进制与十六进制等表示；当整数比较大时，可用下划线作适当分隔。对于浮点数，当数字较大时，可用科学计数法表示。

  Numbers in Python includes integers and floats. Integers can be expressed in binary, octal, or hexadecimal forms. For sufficient large integers, digits can be separated using underscores for grouping. For float numbers, the scientific representations can be adopted for large numbers.

- 当运算中同时有整数与浮点数时，均按浮点数处理然后计算结果。注意，/表示浮点数除法，//表示整除。

  If mixing integers and floats, Python defaults to a float in any operations. Note that / means floating point division and // means integer division.

```
>>> 5 / 2
2.5
>>> 5 // 2
2
>>>
```

其他
Miscellaneous

- Python中没有常量修饰符或关键字，因此常量默认用全部大写变量名表示。

  Python reserves no modifier or keyword for constants, a convention is to use variables of all capital letters for constants.

- 与其它语言的单变量赋值不同，Python可进行多变量赋值：

  Contrasted with other languages, multiple assignment is valid in Python:

  ```
  >>> x, y, z = 0, 0, 0
  ```

- Python中的单行注释，可用#号导引，当注释较多时，用三个引号囊括的多段文本表示。当一行长度过于长不便于阅读需要断行时，用反斜杠表示行的延续。

  In Python, a single line comment is leading by the hash sign (#). For comments that need to span multiple lines, triple quotes are used to delimit the range of comments. For lines of extraordinary length causing problems for readability, backslash is used to wrap the current line of code into next line.

语句
Statements

- 源代码中用于执行的指令称为语句。Python编程语言中有不同类型的语句，如赋值语句、条件语句、循环语句等。当需要进行长计算并且不能将语句放在一行中时，Python 中的语句可以使用圆括号 ()、大括号 {}、方括号 []、分号 (;)、连续字符斜线 (\) 扩展为一行或多行。

  Instructions written in the source code for execution are called statements. There are different types of statements in the Python programming language like Assignment statements, Conditional statements, Looping statements, etc. Multi-Line Statements: When the program needs to do long calculations and cannot fit the statements into one line, it can be extended to one or more lines using parentheses (), braces {}, square brackets [], semi-colon (;), continuation character slash (\).

```
Declared using Continuation Character (\):
s = 1 + 2 + 3 + \
    4 + 5 + 6 + \
    7 + 8 + 9

Declared using parentheses () :
n = (1 * 2 * 3 + 7 + 8 + 9)

Declared using square brackets [] :
footballer = ['MESSI',
          'NEYMAR',
          'SUAREZ']

Declared using braces {} :
x = {1 + 2 + 3 + 4 + 5 + 6 +
     7 + 8 + 9}

Declared using semicolons(;) :
flag = 2; ropes = 3; pole = 4
```

缩进
Indentation

- 块是语句的组合，可被视为用于特定目的的语句分组。大多数编程语言，如 C、C++、Java 都使用大括号 { } 来定义代码块，Python 的显着特点之一是它使用缩进来界定代码块。Python中的缩进用空格来表示，所有缩进距离相同的语句都属于同一个代码块。如果一个块需要更深地嵌套，它只需进一步向右缩进即可。

  A block is a combination of statements. Block can be regarded as the grouping of statements for a specific purpose. Most of the programming languages like C, C++, Java use braces { } to define a block of code. One of the distinctive features of Python is its use of indentation to delimit the blocks of code. Whitespace is used for indentation in Python. All statements with the same distance to the right belong to the same block of code. If a block has to be more deeply nested, it is simply indented further to the right.