Lecture 8 - Python Basic

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

CSC-1004: Computational Laboratory Using Java Course Page: [Click]

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 つへぐ

Comparing Python with Java:

- Java is characterized by its strong typing and performance, making it a common choice for large-scale enterprise applications.
- Python is known for its simplicity and readability, making it popular for rapid development and data analysis.



"Why we need to learn Java programming"

Java has been one of the *most popular programming languages for many years*. It's used in a variety of applications, from web development to mobile apps (especially Android), desktop applications, and large-scale enterprise systems.



"Why we need to learn Python programming"

There's a strong market demand for Python developers in various domains, such as web development, data science, artificial intelligence, and more. Python skills are often sought after by employers.

COMPANIES USING PYTHON





Java example:

3-sum

- Read int values from StdIn.
- Print triples that sum to 0.
- [See Performance lecture]

```
ThreeSum.java
public class ThreeSum
 {
   public static void main(String[] args)
       int N = Integer.parseInt(args[0]);
       int[] a = new int[N]:
       for (int i = 0; i < N; i++)
          a[i] = StdIn.readInt():
       for (int i = 0: i < N: i++)
          for (int j = i+1; j < N; j++)
             for (int k = i+1: k < N: k++)
                if (a[i] + a[j] + a[k] == 0)
                   StdOut.println(a[i] + " " + a[j] + " " + a[k]);
 3
                                                                      (Int
                                                The Chinese University of Hong Kong, Shenzhen
                                              イロト 不得下 イヨト イヨト 二日
```

Python example:

You can *also* write Python code.

Example 2. Use Python like Java.

Noticeable differences

- No braces (indents instead).
- No type declarations.
- Array creation idiom.
- I/O idioms.
- for (iterable) idiom.



% python threesum.py 8 < 8ints.txt	1		
30 -30 0	I .		
30 -20 -10	(深圳)	
-30 -10 40	of Hong K	long,	Shenzhen
-10 0 10	I	_	
	세 문 원	-2	*) Q (*
			2/10

Compiler and Interpreter:

Definition. A compiler translates your entire program to (virtual) machine code.

Definition. An interpreter simulates the operation of a (virtual) machine running your code.







-

2/10

Python Syntax

Some commonly applied Python syntax :

Indentation refers to the spaces (the most common use is four, i.e., <Tab> in your keyboard) at the beginning of a code line. For example:

if 5 > 2:

```
print("Five is greater than two!")
```

• Variables are created when you assign a value to it:

x = 5y = "Hello, World!"



Python Syntax

Some commonly applied Python syntax :

• Casting can be applied to specify the data type of a variable. The type() function

can be used to get the data type of any object.

x = str(3) # x will be '3' y = int(3) # y will be 3 z = float(3) # z will be 3.0print(type(x))

• Comments start with a #, and Python renders the rest of the line as a comment:

```
#This is a comment.
print("Hello, World!")
```



Lists are used to store multiple items in a single variable.

• Lists are created using square brackets:.

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

• To determine how many items a list has, use the len() function:.

print(len(thislist))

• A list can contain different data types:

list1 = ["abc", 34, True, 40.2, "male"]



Lists are used to store multiple items in a single variable.

• List items are indexed and you can access them by referring to the index number:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

• You can specify a range of indexes by specifying where to start and where to end the range.

```
thislist = ["apple", "banana", "cherry", "or-
ange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```



Lists are used to store multiple items in a single variable.

• To determine if a specified item is present in a list use the "in" keyword.

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
```

• To change the value of a specific item, refer to the index number.

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```



Lists are used to store multiple items in a single variable.

• To add an item to the end of the list, use the append() method.

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

• The insert() method inserts an item at the specified index.

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```



イロン イロン イヨン トヨ

Lists are used to store multiple items in a single variable.

• The remove() method removes the specified item.

```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana") # del thislist[1]
print(thislist)
```

• Use the range() and len() functions to create a suitable iterable.

```
thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
    print(thislist[i])
```



Lists are used to store multiple items in a single variable.

• You can loop through the list items by using a while loop.

```
thislist = ["apple", "banana", "cherry"]

i = 0

while i < len(thislist):

print(thislist[i])

i = i + 1
```

• Alternatively, you can loop through the list by using a in.

```
thislist = ["apple", "banana", "cherry"]
for value in thislist:
    print(value)
```



Lists are used to store multiple items in a single variable.

• There are several ways to join, or concatenate, two or more lists in Python.

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]
list3 = list1 + list2
list1.extend(list2)
```



イロン イロン イヨン イヨン 三日

Python supports the usual logical conditions from mathematics:

- Equals: a == b
- Not Equals: a != b
- Less than: a < b
- Less than or equal to: a \leq b
- Greater than: a > b
- Greater than or equal to: a >= b



<ロ> (四) (四) (三) (三) (三)

Python supports the usual logical conditions from mathematics:

- An "if statement" is written by using the if keyword.
- The elif says "if the previous conditions were not true, then try this condition".
- The else keyword catches anything which isn't caught by the preceding conditions.

```
a = 200
b = 33
if h > a
   print("b is greater than a")
elif a == b:
   print("a and b are equal")
else:
   print("a is greater than b")
```



Python supports the usual logical conditions from mathematics:

• The and keyword is used to combine conditional statements:

a = 200 b = 33 c = 500if a > b and c > a: print("Both conditions are True")

• The or keyword is used to combine conditional statements:

```
if a > b or a > c:
    print("At least one of the conditions is
True")
```



Python supports the usual logical conditions from mathematics:

• The not keyword is used to reverse the result of the conditional statement:

a = 33 b = 200 if not a > b: print("a is NOT greater than b")



イロト 不得下 イヨト イヨト 二日

Python Booleans

• Booleans represent one of two values: True or False.

print(10 > 9)print(10 == 9)print(10 < 9)

Almost any value is evaluated to True if it has some sort of content, except 0,

empty strings and other data structures.

bool("abc") bool(123) bool(["apple", "cherry", "banana"])



Python f-strings

f-strings provides a concise and intuitive way to embed expressions and variables directly into strings. The idea behind f-strings is to make string interpolation simpler.

• Print Variables using f-string in Python.

ame = 'Tushar' age = 23 print(f"Hello, My name is {name} and I'm {age} years old.")



Python f-strings

f-strings provides a concise and intuitive way to embed expressions and variables directly into strings. The idea behind f-strings is to make string interpolation simpler.

• Evaluate Expressions with f-Strings in Python.

english = 78 maths = 56 hindi = 85 print(f"Ram got total marks {english + maths + hindi} out of 300")



Python f-strings

f-strings provides a concise and intuitive way to embed expressions and variables directly into strings. The idea behind f-strings is to make string interpolation simpler.

• Printing Dictionaries key-value using f-string in Python.

$$\label{eq:Geek} \begin{split} \mathsf{Geek} &= \{ \ \mathsf{'Id':} \ 112, \ \mathsf{'Name':} \ \mathsf{'Harsh'} \} \\ \mathsf{print}(\mathsf{f}^{"}\mathsf{Id} \ \mathsf{of} \ \{\mathsf{Geek}[\mathsf{'Name'}]\} \ \mathsf{is} \ \{\mathsf{Geek}[\mathsf{'Id'}]\}") \end{split}$$

Dictionaries are used to store data values in key:value pairs.



Python Example Code

Here's an example in Python that incorporates a list, a for loop, if-else statements, and Boolean logic.

```
numbers = [10, 15, 22, 33, 42, 55, 61, 70, 81, 90]
divisor = 5
divisible by divisor, not divisible by divisor = [], []
for number in numbers.
   if number % divisor == 0:
      divisible by divisor.append(number)
   else:
      not divisible by divisor.append(number)
print("Numbers divisible:", divisible by divisor)
                                                                     香港中文大學(深圳)
                                                                     The Chinese University of Hong Kong, Shenzhen
print("Numbers not divisible:", not divisible by divisor)
                                                                   4 D F 4 D F 4 D F 4 D F
```

PyCharm (Download Here)

PyCharm is an integrated development environment (IDE) specifically designed for Python programming, offering code analysis, a graphical debugger, an integrated unit tester, and supports web development with Django.





To create a python project: Step 1 Open PyChram



<ロト < 団ト < 巨ト < 巨ト < 巨ト 三 のへで 9/10

To create a python project: Step 2 Set up a new project



To create a python project: Step 3 done and run the example code





Question and Answering (Q&A)





<ロト < 回 ト < 目 ト < 目 ト 目 の Q () 10 / 10