Lecture 8 - Python Basic

Guiliang Liu

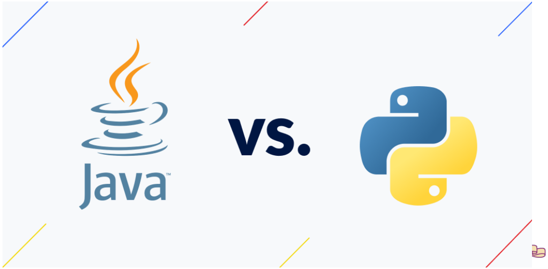
The Chinese University of Hong Kong, Shenzhen

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

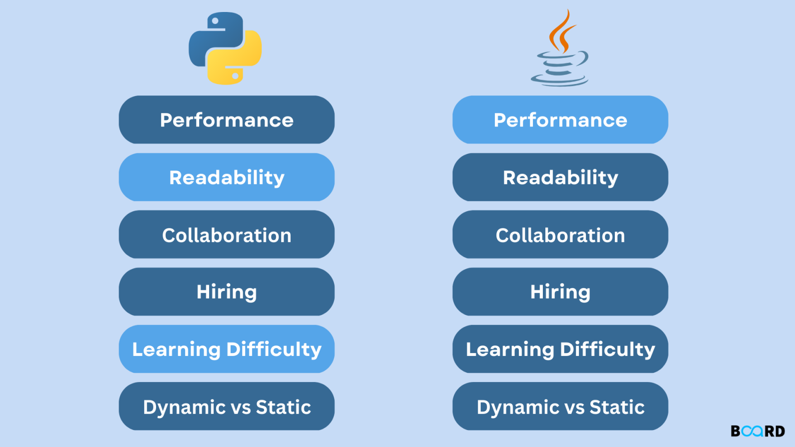
Python v.s. Java

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.

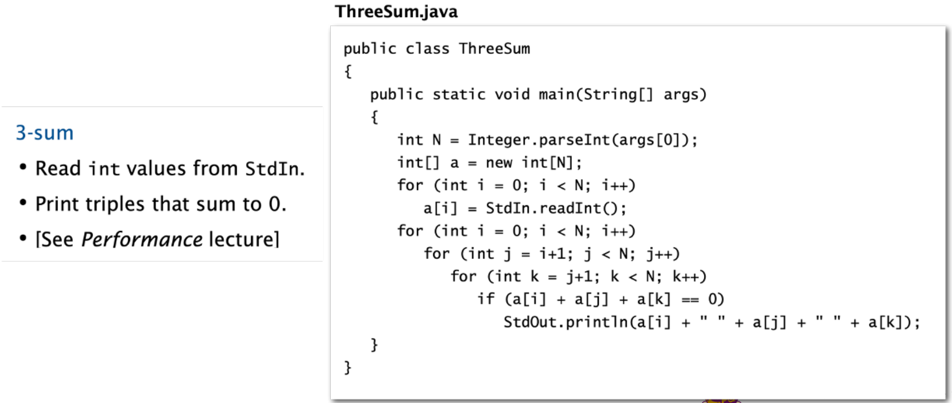


Python v.s. Java



Python v.s. Java

Java example:



3-sum

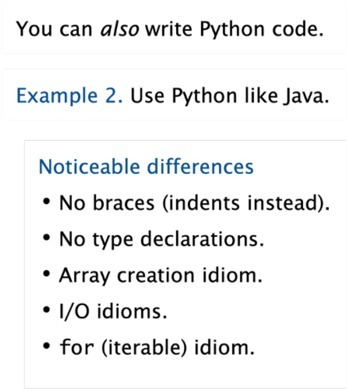
- Read int values from StdIn.
- Print triples that sum to 0.
- [See *Performance* lecture]

ThreeSum.java

```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 = j+1; k < N; k++)
               if (a[i] + a[j] + a[k] == 0)
                  StdOut.println(a[i] + " " + a[j] + " " + a[k]);
   }
}
```

Python v.s. Java

Python example:

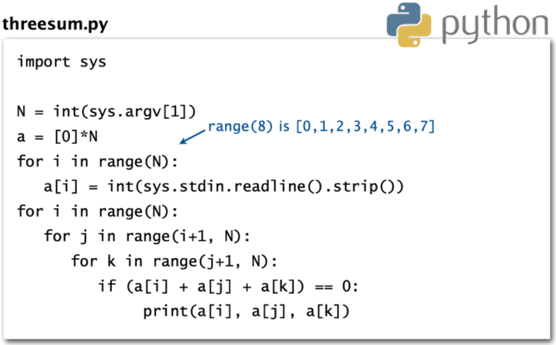


You can *also* write Python code.

Example 2. Use Python like Java.

Noticeable differences

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

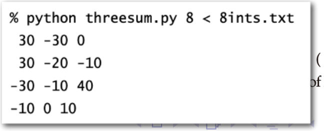


threesum.py

```
import sys

N = int(sys.argv[1])
a = [0]*N
for i in range(N):
    a[i] = int(sys.stdin.readline().strip())
for i in range(N):
    for j in range(i+1, N):
        for k in range(j+1, N):
            if (a[i] + a[j] + a[k]) == 0:
                print(a[i], a[j], a[k])
```

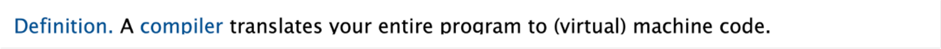
range(8) is [0,1,2,3,4,5,6,7]



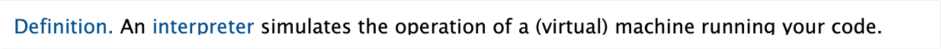
```
% python threesum.py 8 < 8ints.txt
 30 -30 0
 30 -20 -10
-30 -10 40
-10 0 10
```

Python v.s. Java

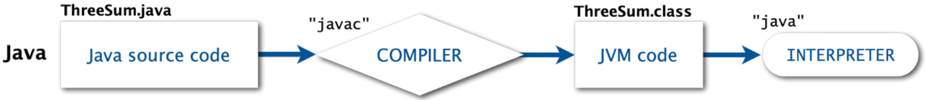
Compiler and Interpreter:

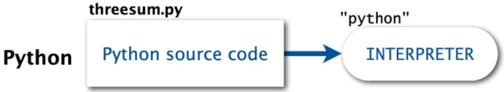


Definition. A compiler translates your entire program to (virtual) machine code.



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





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 = 5
y = "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 3.0
print(type(x))
```

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

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

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Python Lists

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"]
```

Python Lists

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", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

Python Lists

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)
```

Python Lists

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)
```

Python Lists

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")
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])
```

Python Lists

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 range(thislist):
    print(value)
```

Python Lists

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)
```

Python If ... Else

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 <= b
- Greater than: a > b
- Greater than or equal to: a >= b

Python If ... Else

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

Python If ... Else

Python supports the usual logical conditions from mathematics:

- The and keyword is used to combine conditional statements:

```
a = 200
b = 33
c = 500
if 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 If ... Else

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")
```

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 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)
print("Numbers not divisible:", not_divisible_by_divisor)
```

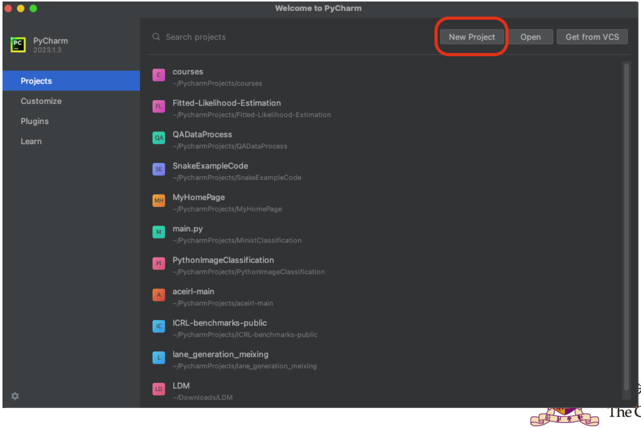
Integrated Development Environment (IDE) for Python

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.

Integrated Development Environment (IDE) for Python

To create a python project: Step 1 pen PyChram



courses
~/PycharmProjects/courses

Fitted-Likelihood-Estimation
~/PycharmProjects/Fitted-Likelihood-Estimation

QADataProcess
~/PycharmProjects/QADataProcess

SnakeExampleCode
~/PycharmProjects/SnakeExampleCode

MyHomePage
~/PycharmProjects/MyHomePage

main.py
~/PycharmProjects/MinistClassification

PythonImageClassification
~/PycharmProjects/PythonImageClassification

aceirl-main
~/PycharmProjects/aceirl-main

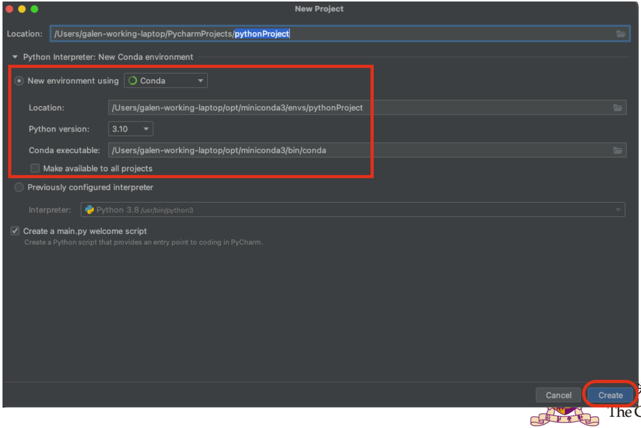
ICRL-benchmarks-public
~/PycharmProjects/ICRL-benchmarks-public

lane_generation_meixing
~/PycharmProjects/lane_generation_meixing

LDM
~/Downloads/LDM

Integrated Development Environment (IDE) for Python

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



New Project

Location: /Users/galen-working-laptop/PycharmProjects/pythonProject

▾ Python Interpreter: New Conda environment

● New environment using Conda

Location: /Users/galen-working-laptop/opt/miniconda3/envs/pythonProject

Python version: 3.10

Conda executable: /Users/galen-working-laptop/opt/miniconda3/bin/conda

□ Make available to all projects

○ Previously configured interpreter

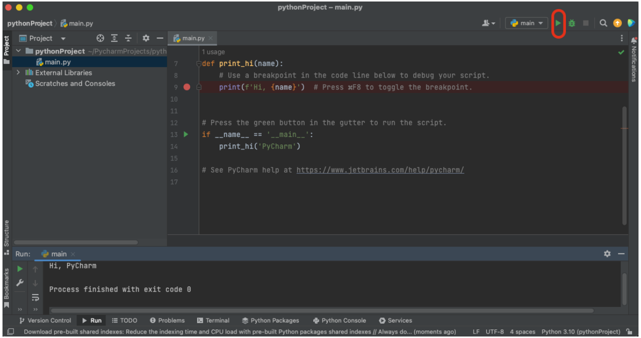
Interpreter: Python 3.8 /usr/bin/python3

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Integrated Development Environment (IDE) for Python

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



```python
def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}')  # Press ⌘F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

```
Hi, PyCharm

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always do... (moments ago)   LF   UTF-8   4 spaces   Python 3.10 (pythonProject)

香 港 中 文 大 學 (深 圳)
The Chinese University of Hong Kong, Shenzhen

Question and Answering (Q&A)