

알고리즘으로 배우는 Python

강사: 오영제

Python 의 역사

- 1991 년 Guido Van Rossum 이 개발 – Open Source
- Web programming 에서 Machine Learning 까지 다양한 분야에 사용
- Python 이름은 snake 가 아니라 drama 제목에서 따옴
- Version 2.7 vs Version 3.8

Why Python ?

Python 은

- 배우기 쉽다
- 비교적 빠르다
- Object Oriented Programming
- Web 에서 Science 까지 사용
- 이식성이 좋다

- C 는 훨씬 빠르지만 배우기 어렵다
- Java 는 속도와 학습 곡선이 C 와 Python 의 중간

Python vs. Java / C++ 선택시 고려 사항

1. 선택기준 : 프로그램 실행 속도 vs. 개발속도 및 유지보수 용이성
 - . 1970/80 – CPU 속도 및 가격이 language 결정 기준
 - . 프로그램 코드의 가독성 및 유지보수 용이성 – shorter is better
2. Python 의 별명은 glue-language
 - . Python 의 Numpy 모듈은 내부적으로 LPACK 사용
 - . Fortran/C/C++ 라이브러리와 호환
 - . Cython 을 이용하여 Python 을 C code 로 변환 가능

Python 은 대부분 task 에서 “fast enough”

*** LPACK (Linear Algebra PACKage with Fortran/C)**



코드길이 비교 – Web Crawler



Java

```
public class CrawlerExample {

    public static void main(String[] args) throws IOException {
        PrintWriter textFile = null;
        try {
            textFile = new PrintWriter("result.txt");
            System.out.println("Enter the URL you wish to crawl..");
            System.out.print("@> ");
            String myUrl = new Scanner(System.in).nextLine();

            String response = getContentByUrl(myUrl);

            Matcher matcher = Pattern
                .compile("href=[\\\"'](?:[^\\"']|')+")
                .matcher(response);
            while (matcher.find()) {
                String url = matcher.group(1);
                System.out.println(url);
                textFile.println(url);
            }
        } finally {
            if(textFile != null) {
                textFile.close();
            }
        }
    }

    private static String getContentByUrl(String myUrl)
        throws IOException {
        URL url = new URL(myUrl);
        URLConnection urlConnection = url.openConnection();
        BufferedReader in = null;
        StringBuilder response = new StringBuilder();
        try {
            in = new BufferedReader(new InputStreamReader
                (urlConnection.getInputStream()));
            String inputLine;
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
        } finally {
            if(in != null) {
                in.close();
            }
        }
        return response.toString();
    }
}
```

입력받은 URL Parsing
및 Web Crawling

Python

```
if __name__ == '__main__':
    with open("result.txt", "wt") as textFile:
        print("Enter the URL you wish to crawl..")
        myUrl = input("@> ")
        for i in re.findall("href=[\\\"'](?:[^\\"']|')+",
            urllib.request.urlopen(myUrl).read().decode(), re.I):
            print(i)
            textFile.write(i + '\n')
```

화면상에서 URL 입력받기

“Hello, World!”

- C

```
#include <stdio.h>
int main(int argc, char ** argv) {
    printf("Hello, World!\n");
}
```
- Java

```
public class Hello{
    public static void main(String argv[]){
        system.out.println("Hello, World!");
    }
}
```
- Python

```
print("Hello, World!")
```

구구단 출력

C++

```
#include "stdafx.h"
#include <iostream>
#include <iomanip>

int main()
{
    int mul = 1;
    int x, y;
    for (x = 2; x <= 9; x++) {
        for (y = 1; y <= 9; y++) {
            mul = x * y;
            cout << setw(1) << x << "*" << y
                << "*" << setw(2) << mul << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Python

```
for x in range(2,10):
    for y in range(1,10):
        mul = x * y
        print("{}*{}={} ".format(x, y, mul),
            end="")
    print()
```

```
2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16
2*9=18 3*1=3 3*2=6 3*3=9 3*4=12 3*5=15 3*6=18 3*7=21
3*8=24 3*9=27 4*1=4 4*2=8 4*3=12 4*4=16 4*5=20
4*6=24 4*7=28 4*8=32 4*9=36 5*1=5 5*2=10 5*3=15
5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45 6*1=6
6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48
6*9=54 7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42
7*7=49 7*8=56 7*9=63 8*1=8 8*2=16 8*3=24 8*4=32
8*5=40 8*6=48 8*7=56 8*8=64 8*9=72 9*1=9 9*2=18
9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

- 2017 카카오 블라인드 공채 코딩 테스트 언어별 통계

- 자바 43%, C++ 36%, 파이썬 11%, 자바스크립트 8%

- 코드 라인 수: C++ > 자바 > 자바스크립트 > 파이썬

- 2019 카카오 블라인드 공채 코딩 테스트 언어별 통계

- 파이썬 60%, 자바 26%, C++ 4%, 자바스크립트 12%

Python 언어의 특징

1. 직관적 자료구조

- String, Decimal, Float, Boolean + List, Tuple, Set, Dictionary

List – array 와 유사 → ['cat', 'dog', 'lion', 'king']

Tuple – list 와 유사하나 값을 변경할 수 없음
→ ('cat', 'dog', 'lion')

Set (집합) → list 와 유사하나 중복을 허용하지 않음

Dictionary – JSON 과 유사 → {'cat': 1, 'dog': 2, 'lion': 3}

2. 영어 문장과 유사한 명령어 형식

- Java: `String name = "Bob";`
`system.out.println(name);`
- C++: `String name = "Bob";`
`cout << name;`
- Python: `name = "Bob"`
`print(name)`

```
for i in range(1, 21):  
    print(i)
```

```
for (int i = 1; i < 21; i++){  
    system.out.println(i)  
}
```

3. One Line Coding (Pythonic Way)

Java: `int temp = x;`
`x = y;`
`y = temp;`

Python: `x, y = y, x`

```
lst1 = []
for m in range(4):
    lst2 = []
    for n in range(4):
        lst2.append(n + m*5)
    lst1.append(lst2)
print(lst1)
```

`[[n + m*5 for n in range(4)] for m in range(4)]`

➔ `[[0, 1, 2, 3], [5, 6, 7, 8], [10, 11, 12, 13], [15, 16, 17, 18]]`

4. Simple Syntax

No { }, No semi-colon / 들여쓰기로 block 구분

```
if a == b:
    c += 1
    print("Same", c)
elif a == d:
    c -= 1
    print("Different", c)
    if final:
        print("final")
    else:
        print("not final")
else:
    return a, b, c
```

5. Dynamically Typed

Java: `int x = 1;`
`x = x / 2;` ➔ `x : 0`

Python: `x = 1`
`x = x / 2` ➔ `x : 0.5`

6. Python 은 Interpreter 언어

- No compile required

7. 풍부한 Libraries (From Web to Data Science)

Python Libraries



8. Supports both Object Oriented and Functional

OOP (Object Oriented Programming)

Class 를 이용한 객체 생성

객체를 통한 method 호출

Functional Programming

함수 (Function) 정의: 함수A => 함수B => 함수C

입력값 처리
출력값 반환

ex) a = [1, 2, 3, 4, 5], b = [2, 4, 6, 8, 10]

`list(map(lambda x, y: x * y, a, b))`

`==> [2, 8, 18, 32, 50]`

9. Everything Simple coding (다양한 함수 내장)

```
>> python -m http-server 8000
```

```
>> from collections import counter
```

```
>> my_list = [1,1, 2, 3, 4, 3, 2, 3, 4, 2, 1, 2, 3, 5]
```

```
>> counter(my_list)
```

```
➔ counter({2: 4, 1: 3, 3: 4, 4: 2, 5: 1})
```

```
>> import itertools
```

```
>> itertools.permutation('ab')
```

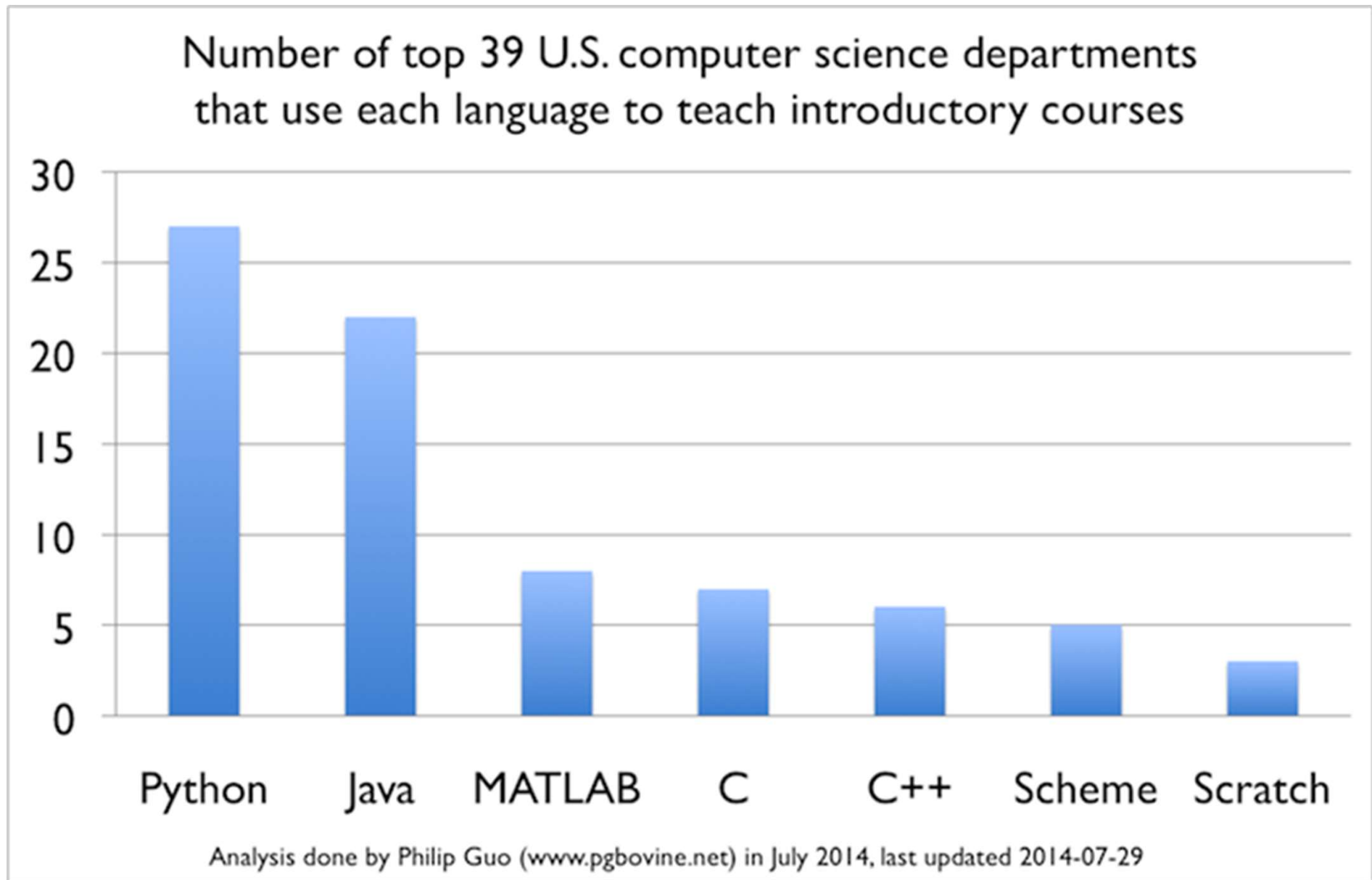
```
➔ [('a', 'b'), ('b', 'a')]
```

```
>> list = [1, 2, 3, 4, 5]
```

```
>> list(reversed(list))
```

```
➔ [5, 4, 3, 2, 1]
```


Computer Science 학과에서 가르치는 First Language



Python 설치 – anaconda (권장)

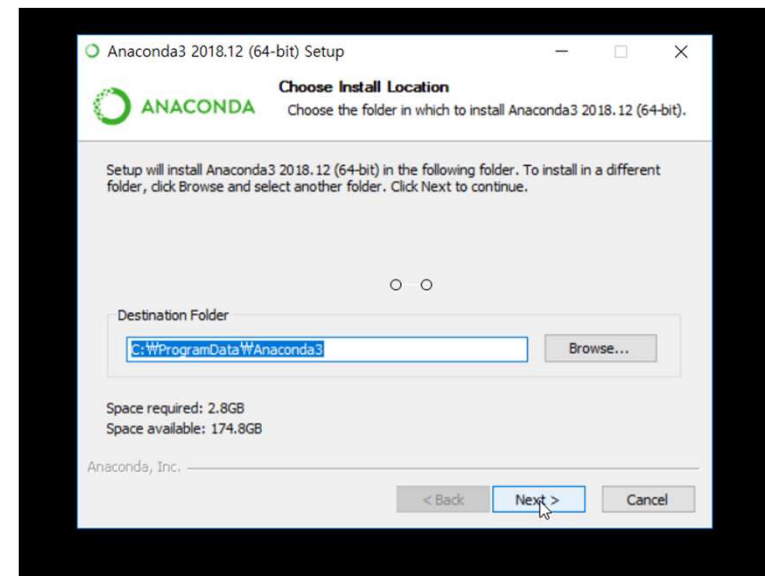
<https://www.anaconda.com/distribution/>

The screenshot shows the Anaconda Python/R Distribution website. The browser's address bar displays [anaconda.com/distribution/#download-section](https://www.anaconda.com/distribution/#download-section). The navigation bar includes links for Windows, macOS, and Linux, with the Windows link highlighted by a red box. The main heading is "Anaconda 2019.10 for Windows Installer". Below this, there are two download options for Windows:

- Python 3.7 version** (highlighted with a red box):
 - Download button
 - 64-Bit Graphical Installer (462 MB)
 - 32-Bit Graphical Installer (410 MB)
- Python 2.7 version**:
 - Download button
 - 64-Bit Graphical Installer (413 MB)
 - 32-Bit Graphical Installer (356 MB)

Below the download options is the section "Get Started with Anaconda Distribution". At the bottom, there is a green cookie consent banner with the text "This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#)" and an "ACCEPT" button. The Windows taskbar is visible at the very bottom.

설치



Python package 설치 (pip)

- 작업 directory 생성: `mkdir xxxx`
- 가상환경 생성 (필요시)
`conda create --name xxxxx python=3.7`
`conda activate xxxxx`
`conda deactivate`
- 패키지 설치 : `pip install package-name`
- 패키지 제거 : `pip uninstall package-name`
- 최신버전 upgrade : `pip install --upgrade package-name`
- 설치된 Python package 확인 : `pip list`

Python 실행

- Anaconda 명령 prompt

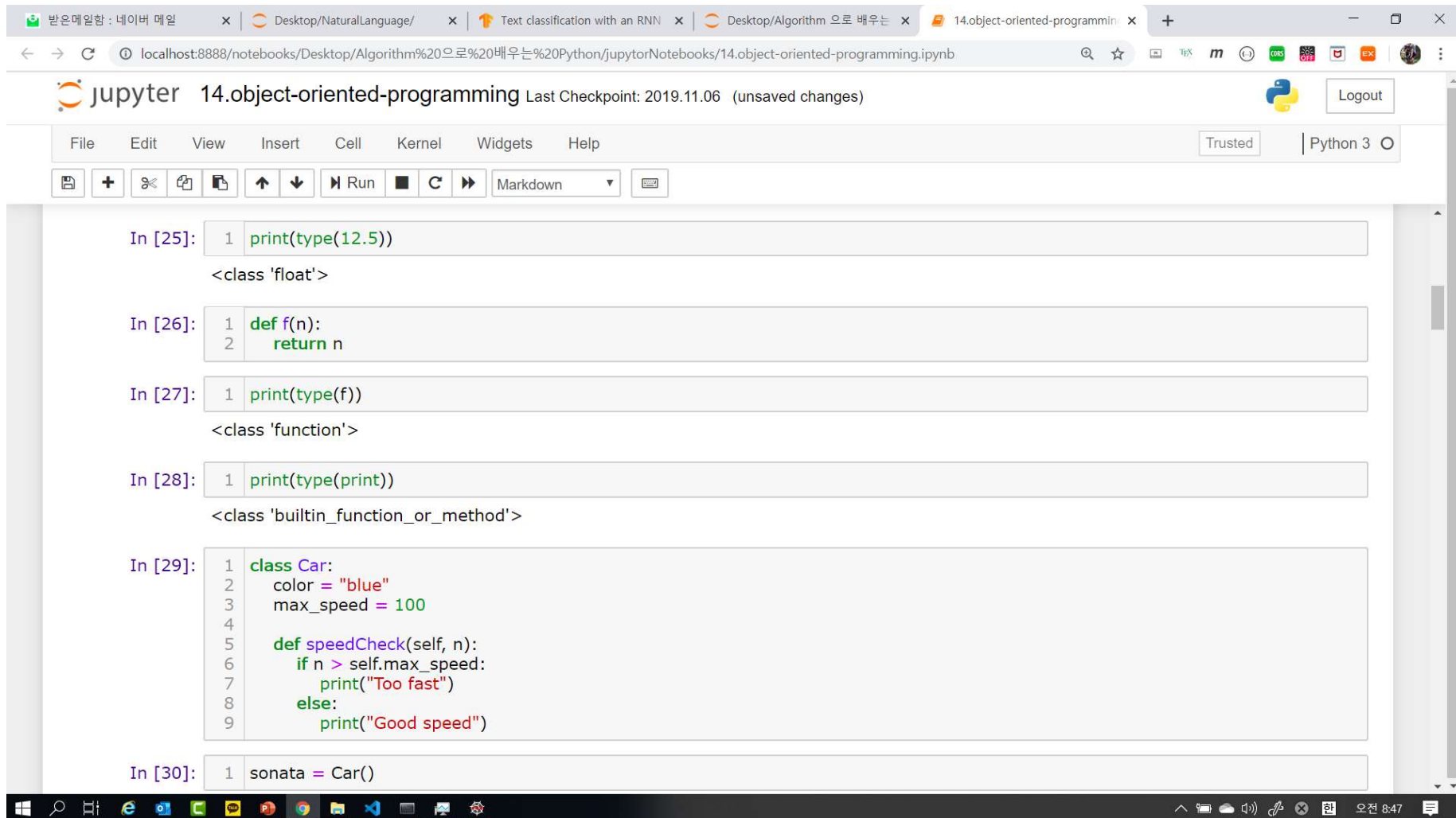
```
(base) C:\Users\trimu>python
```

```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915  
64 bit (AMD64)] :: Anaconda, Inc. on win32
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>>
```

Jupyter Notebook 에서 Python 수행



The screenshot displays a Jupyter Notebook interface in a web browser. The browser's address bar shows the URL: `localhost:8888/notebooks/Desktop/Algorithm%20으로%20배우는%20Python/jupyterNotebooks/14.object-oriented-programming.ipynb`. The notebook's title bar indicates the file name is `14.object-oriented-programming` and shows the last checkpoint was on 2019.11.06 with unsaved changes. A 'Logout' button is visible in the top right corner. The interface includes a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for saving, opening, and running code. The main area contains six code cells, each with a prompt (e.g., 'In [25]:') and its output. The code cells contain the following Python code:

```
In [25]: 1 print(type(12.5))
          <class 'float'>

In [26]: 1 def f(n):
          2     return n

In [27]: 1 print(type(f))
          <class 'function'>

In [28]: 1 print(type(print))
          <class 'builtin_function_or_method'>

In [29]: 1 class Car:
          2     color = "blue"
          3     max_speed = 100
          4
          5     def speedCheck(self, n):
          6         if n > self.max_speed:
          7             print("Too fast")
          8         else:
          9             print("Good speed")

In [30]: 1 sonata = Car()
```

The Windows taskbar at the bottom shows the system clock as 8:47 AM on October 1, 2019.

SPYDER

The image shows the Spyder Python IDE interface. The main window has a menu bar (File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help) and a toolbar. The file explorer shows a project named 'temp.py' located at 'C:\Users\Wtrimu\temp.py'. The code editor displays a Python function named 'flatten_dict' with the following code:

```
1 def flatten_dict(adict, parent_key=None, flatten_result=None):
2
3     if flatten_result is None:
4         flatten_result = {}
5
6     #print("adict=", adict)
7
8     for k, v in adict.items():
9         if isinstance(v, dict):
10             flatten_dict(v, k, flatten_result)
11         else:
12             if parent_key:
13                 flatten_result[parent_key + '.' + k] = v
14             else:
15                 flatten_result[k] = v
16                 #print("flatten_result =", flatten_result)
17
18     return flatten_result
19
20 flatten_dict({'a': 1, 'b': {'x': 2, 'y': 3}, 'c': 4})
```

The console panel on the right shows the IPython prompt and the output of the function call:

```
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.11.1 -- An enhanced Interactive Python.

In [1]:
```

The help panel on the right shows the 'Usage' section, which explains how to get help for any object by pressing Ctrl+I in front of it, either on the Editor or the Console. It also mentions that help can be shown automatically after writing a left parenthesis next to an object. A link to the tutorial is provided: [New to Spyder? Read our tutorial](#).

The status bar at the bottom shows the current file is 'temp.py', the console is active, and the history is shown. The status bar also displays the current line and column (Line 20, Col 1), the encoding (ASCII), the line ending (CRLF), the read/write status (RW), and the memory usage (Mem 38%).

기타

- PyCharm
- atom
- Sublimetext
- Visual Studio Code, etc

Github Repository

<https://github.com/ironmanciti/algorithmPython>

ironmanciti / algorithmPython

Watch 0 Star 1 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

11 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file **Clone or download**

ironmanciti 'nov4' Latest commit f4f4ae8 14 hours ago

.ipynb_checkpoints	'nov4'	14 hours ago
.vscode	'nov4'	14 hours ago
babynames	'nov4'	14 hours ago
README.md	first commit	7 months ago
Review First Day.ipynb	pythonb	7 months ago
Review Second Day-Copy1.ipynb	pythonb	7 months ago
Review Second Day.ipynb	pythonb	7 months ago
Review Third Day.ipynb	'nov4'	14 hours ago
Review Third Day.py	3rd	7 months ago