



# 2018 Asia University AI Summer Program (Day3-July5) A: Python Basics



Frank Hsueh-Ting Chu

# Content

- Anaconda
- Jupyter notebook
- Markdown
- Python Basics: variables, statements
- The Python Standard Library
- Python ML Libraries: numpy scipy matplotlib




# Anaconda

 Download Git(Git-2.18.0-64-bit.exe)

 Download Anaconda 5.2(Anaconda3-5.2.0-Windows-x86\_64.exe)

 Download Visual Studio Code1.24(VSCodeSetup-x64-1.24.1.exe)

 Download cuda toolkit 9.0(cuda\_9.0.176\_win10.exe)

 Download cudnn(cudnn-9.0-windows10-x64-v7.zip)

 Download Voice Kit SD image(aiyprojects-2018-04-13.img.xz)

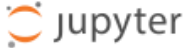
 Download card writing utility Etcher.io (Etcher-Setup-1.4.4-x64.exe)

 Download putty.exe

 Download Day3 Notebooks(ml.zip)



# Jupyter notebook

 Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ▾ ↺

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📄	day2b.ipynb	18 hours ago	
<input type="checkbox"/>	📄	day3a.ipynb	a minute ago	
<input type="checkbox"/>	📄	day3b.ipynb	a minute ago	
<input type="checkbox"/>	📄	day3c.ipynb	a minute ago	
<input type="checkbox"/>	📄	day3d.ipynb	a minute ago	
<input type="checkbox"/>	📄	day3e.ipynb	a minute ago	
<input type="checkbox"/>	📄	ex1.py	19 hours ago	
<input type="checkbox"/>	📄	ex2.py	19 hours ago	

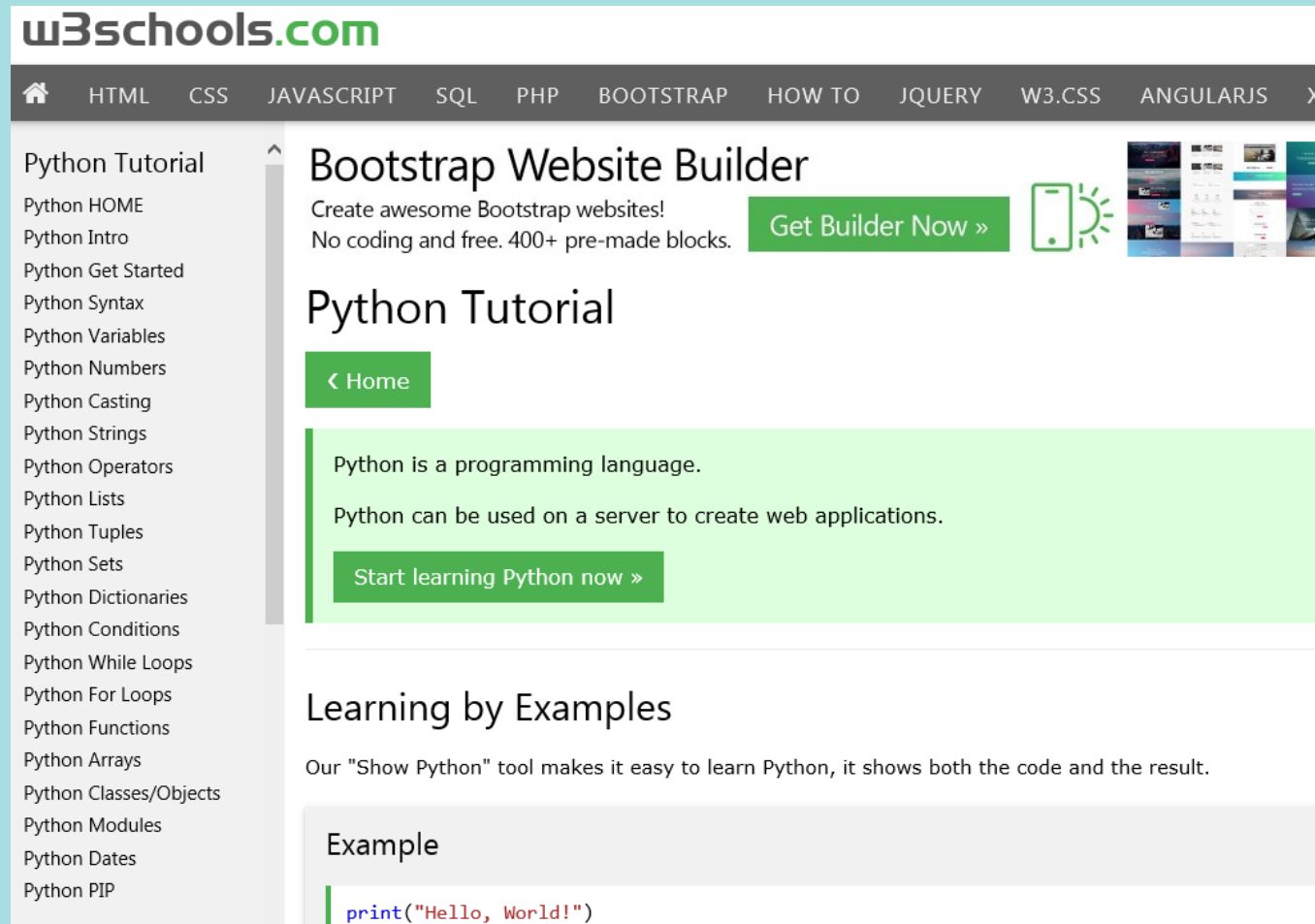


# Markdown

- Headings
  - #
  - ##
- Links
  - `http://xxx.yyy.zzz`
- Lists:
  - 1. aaa
  - \* bbb



# Python Basics: variables, statements



The screenshot shows the w3schools.com website. The top navigation bar includes links for HTML, CSS, JAVASCRIPT, SQL, PHP, BOOTSTRAP, HOW TO, JQUERY, W3.CSS, ANGULARJS, and XML. On the left, a sidebar lists various Python topics under the heading "Python Tutorial". The main content area features a "Bootstrap Website Builder" advertisement, a "Python Tutorial" section with a "Home" button, and a "Learning by Examples" section. The "Learning by Examples" section includes a code example: 

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

w3schools.com

HTML CSS JAVASCRIPT SQL PHP BOOTSTRAP HOW TO JQUERY W3.CSS ANGULARJS XML

Python Tutorial

- Python HOME
- Python Intro
- Python Get Started
- Python Syntax
- Python Variables
- Python Numbers
- Python Casting
- Python Strings
- Python Operators
- Python Lists
- Python Tuples
- Python Sets
- Python Dictionaries
- Python Conditions
- Python While Loops
- Python For Loops
- Python Functions
- Python Arrays
- Python Classes/Objects
- Python Modules
- Python Dates
- Python PIP

**Bootstrap Website Builder**

Create awesome Bootstrap websites!  
No coding and free. 400+ pre-made blocks.

[Get Builder Now »](#)

**Python Tutorial**

[< Home](#)

Python is a programming language.

Python can be used on a server to create web applications.

[Start learning Python now »](#)

**Learning by Examples**

Our "Show Python" tool makes it easy to learn Python, it shows both the code and the result.

**Example**

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



# The Python Standard Library

## The Python Standard Library

While [The Python Language Reference](#) describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the [Python Package Index](#).

- 1. Introduction
- 2. Built-in Functions
- 3. Built-in Constants
  - 3.1. Constants added by the `site` module
- 4. Built-in Types
  - 4.1. Truth Value Testing
  - 4.2. Boolean Operations — `and`, `or`, `not`
  - 4.3. Comparisons
  - 4.4. Numeric Types — `int`, `float`, `complex`
  - 4.5. Iterator Types
  - 4.6. Sequence Types — `list`, `tuple`, `range`
  - 4.7. Text Sequence Type — `str`
  - 4.8. Binary Sequence Types — `bytes`, `bytearray`, `memoryview`
  - 4.9. Set Types — `set`, `frozenset`
  - 4.10. Mapping Types — `dict`
  - 4.11. Context Manager Types
  - 4.12. Other Built-in Types
  - 4.13. Special Attributes
- 5. Built-in Exceptions

## 11. File and Directory Access

- 11.1. `pathlib` — Object-oriented filesystem paths
- 11.2. `os.path` — Common pathname manipulations
- 11.3. `fileinput` — Iterate over lines from multiple input streams
- 11.4. `stat` — Interpreting `stat()` results
- 11.5. `filecmp` — File and Directory Comparisons
- 11.6. `tempfile` — Generate temporary files and directories
- 11.7. `glob` — Unix style pathname pattern expansion
- 11.8. `fnmatch` — Unix filename pattern matching
- 11.9. `linecache` — Random access to text lines
- 11.10. `shutil` — High-level file operations
- 11.11. `macpath` — Mac OS 9 path manipulation functions

## 12. Data Persistence

- 12.1. `pickle` — Python object serialization
- 12.2. `copyreg` — Register `pickle` support functions
- 12.3. `shelve` — Python object persistence
- 12.4. `marshal` — Internal Python object serialization
- 12.5. `dbm` — Interfaces to Unix "databases"
- 12.6. `sqlite3` — DB-API 2.0 interface for SQLite databases

## 13. Data Compression and Archiving

- 13.1. `zlib` — Compression compatible with **gzip**
- 13.2. `gzip` — Support for **gzip** files
- 13.3. `bz2` — Support for **bzip2** compression
- 13.4. `lzma` — Compression using the LZMA algorithm
- 13.5. `zipfile` — Work with ZIP archives
- 13.6. `tarfile` — Read and write tar archive files

## 14. File Formats

- 14.1. `csv` — CSV File Reading and Writing
- 14.2. `configparser` — Configuration file parser
- 14.3. `netrc` — `netrc` file processing
- 14.4. `xdrlib` — Encode and decode XDR data
- 14.5. `plistlib` — Generate and parse Mac OS X `.plist` files

## 15. Cryptographic Services

- 15.1. `hashlib` — Secure hashes and message digests
- 15.2. `hmac` — Keyed-Hashing for Message Authentication
- 15.3. `secrets` — Generate secure random numbers for managing secrets

## 16. Generic Operating System Services

- 16.1. `os` — Miscellaneous operating system interfaces
- 16.2. `io` — Core tools for working with streams
- 16.3. `time` — Time access and conversions

# A simple crawler

```
import urllib.request  
import re
```

```
with urllib.request.urlopen('http://www.asia.edu.tw/news1.php') as response:  
    html = response.read().decode('utf-8')  
    pattern = '<font color="#446666" face="微軟正黑體" size="2">'  
    for pos in re.finditer(pattern,html):  
        pos2 = html.find('</font>', pos.end())  
        sub = html[pos.end():pos2]  
        print(sub)
```







Thanks!

Q&A

