

Topic 1 Introduction to Data Programming

Learning Outcomes

After completing this topic and the recommended reading, you should be able to:

- Set up and run Jupyter Notebook on a Windows, Mac or Linux operating system.
- Use Jupyter Notebook to write and edit code.
- Write and explain simple Python programs using variables and mathematical operators.

1. Introduction to Data Programming

Data (definition)

• "Facts and statistics collected together for reference or analysis."

[Oxford English Dictionary]

 "Information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer."

[Cambridge Dictionary]

• "Factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation."

[Merriam-Webster]

• "Data are individual facts, statistics, or items of information, often numeric, that are collected through observation. In a more technical sense, data are a set of values of qualitative or quantitative variables about one or more persons or objects."

[Wikipedia]

Information (definition)

• "Facts provided or learned about something or someone."

[Oxford English Dictionary]

• "Facts or details about a situation, person, event, etc."

[Cambridge Dictionary]

• "Knowledge obtained from investigation, study, or instruction."

[Merriam-Webster]

• "Knowledge communicated or received concerning a particular fact or circumstance; knowledge gained through study, communication, research, instruction, etc."

[Dictionary.com]

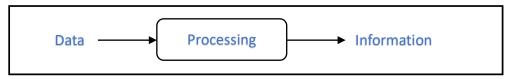
Data vs. Information

- Data
 - o Raw, unorganised facts that need to be processed.
 - Unusable until it is organised.

• Information

- o Created when data is processed, organised, and structured.
- Needs to be put in an appropriate context in order to become useful.

Data Science



Programming and Data

- Tasks to undertake for data programming
 - Data collection
 - Data processing (wrangling)
 - Data visualisation

Train and apply algorithms from fields such as machine learning,
 statistics, data mining, optimisation, image processing, etc.

• Programming

- The process of producing an executable computer program that performs a specific task.
- The purpose is to find a sequence of instructions that automate the implementation of the task for solving a given problem.

• Programming Language

- The source code of a program is written in one or more languages that are intelligible to humans, rather than machine code, which is directly executed by the CPU.
- o Python ₱ python

 o Py
 - https://www.python.org/

2. Introduction to Development Environments

Source-code Editors

- **Source-code editor**, or programming text editor, is a fundamental programming tools designed specifically for editing source code of computer programs.
- It highlights the syntax elements of your programs, and provides many features that aid in your program development.
- Examples:
 - Visual Studio Code [https://code.visualstudio.com/]
 - Notepad++ (Windows only) [<u>https://notepad-plus-plus.org/]</u>
 - o Vim [https://www.vim.org/]
 - Sublime Text (not open source) [<u>https://www.sublimetext.com/</u>]
 - Atom [<u>https://atom.io/</u>]
 - o Emacs [https://www.gnu.org/software/emacs/]
 - TextMate (Macs only) [https://macromates.com/]
 - Jupyter Jupyte
 - https://jupyter.org/

Integrated Development Environments (IDEs)

- Integrated development environment is a software application that
 provides comprehensive facilities to computer programmers for software
 development.
- An IDE normally consists of at least a <u>source code editor</u>, <u>build</u> <u>automation tools</u> and a <u>debugger</u>.
- Examples:
 - o Spyder [https://www.spyder-ide.org/]

- o RStudio [https://rstudio.com]
- Eclipse [<u>https://www.eclipse.org/</u>]
- Microsoft Visual Studio [<u>https://visualstudio.microsoft.com/vs/</u>]
- Wing Python IDE [<u>https://wingware.com</u>]

Markdown / Markup Languages

- *Markdown* is a markup language that consists of a set of rules for adding formatting elements to plain text documents
 - Boldface, italics, headers, paragraphs, lists, code blocks, images, etc.
 - o https://www.markdownguide.org/
- Invented by *John Gruber*
 - The overriding design goal for Markdown's formatting syntax is to make it as readable as possible.
 - The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions
- Examples
 - HTML; XML; LaTeX

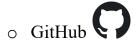
Version Control Systems

- *Version Control* is a class of systems responsible for managing changes to computer programs, documents, large websites, or other collections of information.
- *Version Control Systems* (VCS) are software tools that help software teams manage changes to source code over time.
 - Undertakes the tedious task of keeping track of the changes to all project's files and who made them

- o Allows users to recover any previous version at any given time
- Examples:
 - Subversion [https://subversion.apache.org]



https://git-scm.com/



https://github.com/

Package/Environment Manager

- *Package manager*, or package management system, is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer in a consistent manner. Also deals with packages, distributions of software and data in archive files.
- *Environment manager* enables personalised, consistent desktop environments without cumbersome roaming profiles or scripts.
- Example:
 - Anaconda
 - https://www.anaconda.com/

Installing Anaconda

- Go to Anaconda, download Anaconda Individual Edition
 - o https://www.anaconda.com/products/distribution
- Packages include
 - conda
 - package management system

- o pandas, scikit-learn, nltk
 - packages for data science
- Anaconda Navigator
 - a graphical user interface
- o QtConsole
 - an interactive Python environment
- Spyder
 - a standard cross-platform IDE for Python
- Jupyter Notebook
 - an interactive web-browser based application for creating and sharing code

Package Installer for Python (pip)

- *pip* is the de facto and recommended package-management system written in Python and is used to install and manage software packages.
- It connects to an online repository of public packages, called the Python Package Index.
- We use *pip* to install packages from the Python Package Index
- Examples
 - o pip install beautifulsoup4
 - o pip install -r dependencies
 - Install packages based on dependencies in code
 - o pip freeze
 - See all the packages installed

3. Introduction to Python

- Open-source, interpreted, high-level, object-oriented, general-purpose, easy to download, write and read
- Named for the British comedy group *Monty Python*
- Simpler language, allow us to focus less on the language and more on problem solving
- Many of the best parts of other languages are included
 - Data structures
 - o Controls
 - Many packages for common tasks

Variables

- *Variable* is a named piece of memory whose value <u>can change</u> during the running of the program; *constant* is a value which cannot change as the program runs.
 - Python doesn't use constant
- We use *variable names* to represent objects (number, data structures, functions, etc.) in our program, to make our program more readable.
 - o All variable names must be one word, spaces are never allowed.
 - Can only contain alpha-numeric characters and underscores.
 - o Must start with a letter or the underscores character.
 - o Cannot begin with a number.
 - o Case-sensitive
 - o Standard way for most things named in Python is <u>lower with under</u>
 - Lower case with separate words joined by an underscore

Comments

- Not processed by the computer, valued by other programmers.
- Header comments
 - o Appear at beginning of a program or a module
 - o Provide general information
- Step comments or in-line comments
 - Appear throughout program
 - o Explain the purpose of specific portion of code
- Often comments delineated by
 - // comment goes here
 - o /* comment goes here */
 - # Python uses this

Python Operations

- Assignment Operator
 - o "="
 - o Example:
 - a = 67890/12345
 # compute the ratio, store the result in ram, assign to a
 # the value of a is 5.499392
 - b = a# b pointing to value of a
- Output
 - o "print()"
 - o Example:
 - print('Hello World!') # print the string literals
 - print(a) # print the value of a

Data Types in Python

- Declaration of variables in Python is not needed
 - Use an assignment statement to create a variable

• Float

- Stores real numbers
- o a = 4.6
- o print(type(a))

• Integer

- Stores integers
- b = 10
- o print(type(b))

• Conversion

 \circ int(a) # convert float to int => 4 \circ float(b) # convert int to float => 10.0

• Basic arithmetic operators

0 10 // 3

Floor Division

=> 3

• String

- o Stores strings
- o phrase = 'All models are wrong, but some are useful.'

o *phrase[0:3]*

slicing character 0 up to 2

=> All

o phrase.find('models')

find the starting index of word

=> 4

o phrase.find('right')

word not found

=> -1

o phrase.lower()

set to lower case

=> 'all models are wrong, but

some are useful.'

o phrase.upper()

set to upper case

=> 'ALL MODELS ARE

WRONG, BUT SOME ARE

USEFUL.'

o phrase.split(', ')

split strings into list, base on delimiter

=> ['All models are wrong',

'but some are useful.']

Boolean

- O Stores logical or <u>Boolean</u> values of <u>TRUE</u> or <u>FALSE</u>
- o k = 1 > 3
- o print(k)

False

o print(type(k))

```
<class 'bool'>
```

Logical operators

- o Conjunction (AND): "and"
- o Disjunction (OR): "or"
- o Negation (NOT): "not"

<u>a</u>	<u>b</u>	a and b	<u>a or b</u>	not a
T	Т	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

Data Structures in Python

• Tuples

- o Store ordered collection of objects
- o Immutable: elements cannot be modified, added or deleted
- o Written with round brackets "()"
 - tuple1 = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
 - tuple2 = ("Handsome Koh", 4896, 13.14, True)
- o Accessing elements by indexing
 - tuple1[0] #first element index => 'apple'
 - tuple1[-1] # last element index => 'mango'

• Lists

- o Store ordered collection of objects; mutable
- Written with square brackets "[]"
 - list1 = ["apple", "banana", "cherry"]
 - *list2* = ["Handsome Koh", 4896, 13.14, True]
- Changing elements

 - list1.remove("apple") # delete elements
 => ['banana', 'coconut', 'orange']

Sets

- o Store unordered, unindexed, nonduplicates collection of objects
- O Written with square brackets "{ }"
 - *set1* = { "apple", "banana", "cherry"}
 - *set2* = { "apple", "samsung"}
- Set operations

 - set1.intersection(set2) # Intersect both sets
 => {'apple'}

Dictionaries

- o Store unordered collection of objects
- Written with square brackets "{ }", and "key:value" pair
 - thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}
- o Accessing/modifying elements by key name

4. Introduction to Jupyter Notebook

- **Jupyter Notebook** is a web-based interactive computing platform.
- "Julia" + "Python" + "R"
- Integrate code and output into a single document contains:
 - <u>Live code</u>, <u>mathematical equations</u>, <u>visualisations</u>, and
 explanatory/narrative text, interactive dashboards and other media
- Can be easily shared
 - o Notebook files have ".ipynb" extension
 - o Export to ".html" and ".pdf" extensions
- Launch "Jupyter Notebook" from "Anaconda Navigator"
- Create new notebook
 - o "File" → "New Notebook" → "Python 3"
- Exporting notebook
 - o "File" → "Download as" → "HTML (.html)"
 - o "File" → "Print Preview" (for PDF)
- Shutting Down Jupyter
 - o "File" → "Close and Halt"
 - o Quit

5. Exercises

1.301 Practice Exercises (Coursera)

• Refers to "1.301 part-1.html"

1.302 A bit more Python – our first downloadable notebook! (Coursera)

• Refers to "1.302 pythonPractice.html"

1.304 World's Population

• Refers to "1.304 Topic 1 - Lab.html"

6. Practice Quiz

• Work on *Practice Quiz 01* posted on Canvas.

Useful Resources

•

o <u>http://</u>