The Health Informaticist

Hants Williams

2024-10-28

Table of contents

Preface

Why?

I started off, and still identify as a nurse, but now find myself somewhere between the interaction of clinician and technologist. My job as a professor involves teaching, and constantly aquiring new knowledge, abilities, and skills (KSAs) between evidence based medicine (EBM) and technology-focused tools that could be used to help us address these issues. My curiosity and passion for BOTH healthcare and technology has led me to pursue a variety of topics that I try to address in this book. So what does this book cover?

This book covers python, healthcare data, medical codexes, open source datasets, databases, cloud technologies, inferential statistics and machine learning, visualizations, and related technologies important to understand as a modern (future) health informaticsts.

I've created this book as the most important topics to me, and what I have experienced within academic medical hospital systems, private hospitals, consulting, and health tech-startups. So whether you're a healthcare professional, data scientist, student, or enthusiast, this book will offer you valuable insights and hopefully fun conversation and dialgue related to what can be dry and boring material.

What You'll Find Here:

- **A Broad Foundation**: Rather than in-depth on each topic, this book provides a wide overview, giving you a base to explore further.
  - An intro to…*Python*
  - An intro to…*Healthcare Data*
  - An intro to…*Inferential Statistics*
  - An intro to…*Ai and Machine Learning*
  - An intro to…*Supporting Cloud Technologies*
- **Hands-On Learning**: Chapters combine theory and practical examples, allowing you to apply what you learn through Pyodide-powered, interactive Python exercises.

Think Legos...

Think of each section as a component, or a lego, that when combined together can lead to the creation of something useful. These core technologies: databases, scripts, clouds, code languages, databases, medical codexes...etc that we will be exploring should be viewed as individual lego pieces. It is our job as health informaticists to understand which pieces exist, what they are capability of providing to us, and then how we can put them together to make something unique and useful. Key word is useful.

Figure 1: Image of LEGO bricks as a metaphor for learning blocks

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

2 Python Code - 1 - Hello World

2.1 Objectives

The aim of this exercise is to learn how to write and execute Python code.

2.2 Rationale

There are plenty of programing languages, but all of them share some common principles. Computers execute machine code, a binary set of instructions that are carried out by the computer processors, but usually human do not write binary machine code, or even assembly language, the written code directly translatable to machine code.

Most of the time programers write code in languages that are easier to write and read by us, humans. These codes, in turn, have to be translated to machine code. This process of translation is called compilation. Compilation is a complex process that comprises several steps, and different compilers and programming languages divide theses steps differently. There are compilers, like the C or C++ compiles, that take the code and compile it into a binary executable that can be directly run in a computer, while there are other, like the Python interpreter of the Java compiler, that are capable of running directly the human readable code that create an intermediate representation called bytecode. In these cases we need to install the language interpreter to run the program. For instance, if you want to run a Java or Python program you will need to install Java or Python before.

Python is an interpreted language, so, once you install Python you will be able to directly run Python code. However, you first need to have Python available in your computer.

There are many ways of installing Python like:

- Downloading Python from python.org.
- Using the Windows store or the Linux packages already prepared.
- Using a Python version manager like uv.
- Having Python included in your web browser thanks to Pyodide.

In this exercise we are going to use Pyodide, a Python version that allow us to run Python code directly in a web page without requiring any previous installation.

The Notebook document is composed by cells, some are text (in markdown format), and others contain Python code that can be executed. To execute a Python cell:

- Select the cell that you want to execute.
- Click the play symbol on top of the page or press shift + enter.

2.3 Print Hello world

```
#| exercise: hello_world_01
print("Hello world")
```

2.4 Print Hello "Your Name"

```
#| exercise: hello_world_02
# Use this Notebook cell to write Python code capable of printing your name
```

Solution.

Tip

```
print('Jane')
```

3 Python Code - 2 - Flow and Variables

3.1 Resources

- Variables in Python in Real Python.
- print function official documentation.
- A print function tutorial in Real Python.
- Variable unpaking tutorial in Real Python.

3.2 Flow

The computer executes the programming code one line (or statement) at a time. The order in which the lines are executed is called flow.

```
print("This line will be executed first.")
print("This line will be executed second.")
print("This line will be executed last.")
```

Order the lines in the following code, we want the computer to first say Hello, then Your Name, and, finally, an invitation to play.

```
#| exercise: fix_flow
print("Your Name")
print("Do you want to play a nice game of chess?")

print("Hello")
```

Note

Remember that the lines are executed in order, so change the order of the lines.

Solution.