Lecture 1: Introduction

Morten Rieger Hannemose, Vedrana Andersen Dahl

Fall 2023

What is a program?

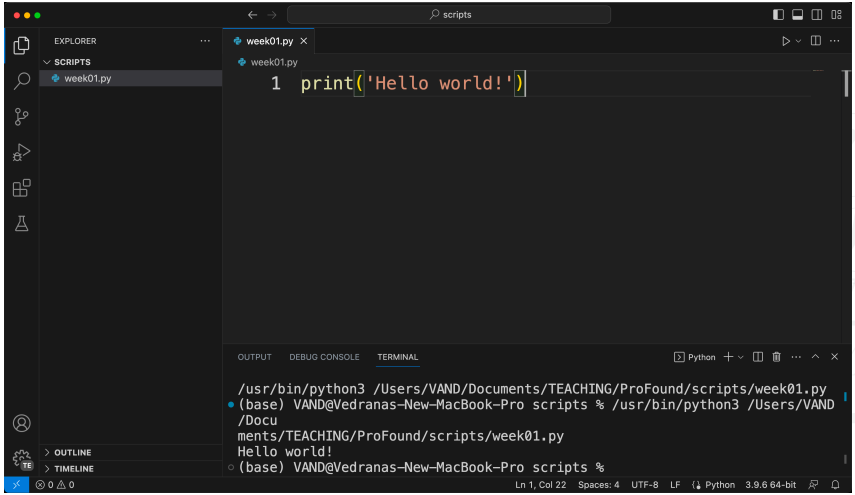
A program

- Instructions specifying actions or computations to be performed by *the computer* (as if computer was a person)
- Analogy to recipes: structure, commands, ordering, level of detail, outcome...
- Why write programs?
- How to learn programming?

Programming language

- Artificial (formal) language for giving instructions to *the computer*
- Python language (and programming languages in general) is purely textual
- Syntax - a set of rules for a valid program (grammar)
- Words (tokens) in a programming language: keywords, variable, functions...

Hello world!


scripts
EXPLORER
SCRIPTS
week01.py
week01.py
1 print('Hello world!')
OUTPUT
DEBUG CONSOLE
TERMINAL
Python
/usr/bin/python3 /Users/VAND/Documents/TEACHING/ProFound/scripts/week01.py
(base) VAND@Vedranas-New-MacBook-Pro scripts % /usr/bin/python3 /Users/VAND/Docu
ments/TEACHING/ProFound/scripts/week01.py
Hello world!
(base) VAND@Vedranas-New-MacBook-Pro scripts %
OUTLINE
TIMELINE
Ln 1, Col 22
Spaces: 4
UTF-8
LF
Python
3.9.6 64-bit

Assignments

- Assignment statement:

  variable name (LEFT) = something that can be evaluated (RIGHT)

- Assigns a value to a variable; If needed, creates a variable
- Assignment is an instruction, code that has an effect
- How to access values? Variable names! After the assignment, we can use variable names to access the values
- Analogy to assigning a phone number to contacts, or assigning a document to a file name
- Assignment allows overwriting!
- Allowed variable names? Rules and conventions
- Values?

Getting started with assignments

Simple types (of values)

- We start with 3 simple types:
  - integers (int)
  - floating-point numbers (float)
  - strings (str)

Expressions (something that can be evaluated)

- We start with simple expressions:
  - Binary operators, like $5 + 9.2$
    Operators have different effects depending on the variable types
  - Sneak peak: unary functions, like `math.sin(0.5)`. In the book, this comes a bit later (3.2) as it requires math package, and packages come later

How are programs executed?

- Simple flow of execution:
  - Line-by-line from top to bottom

Examples

Example 1

```
x = 8
y = 13
y = x
print(x)
x = y
print(x)
```

Example 2

```
this = 200
print(this)
this = this + 21
print(this)
```

Example 3

```
# Compute the area of the rectangle
length = 5  # The length
width = 3   # The width

# Calculate the area using assignment
area = length * width

# Display the result
print("The area is:", area)
```

Example 4

```
one = "one"
two = "two"
three = one + two
print(three)
```

Getting started with programming

How do we execute programs?

- Simple (minimalistic) style
  - Terminal, running commands (interactive)
  - Terminal, starting the Python shell
  - Terminal, running a Python file
- Many other options, including running code from the browser

How do we write programs?

- Source-code editor: VSCode, Spyder
- Functionality: Syntax check, debugger, variable explorer. . . and many many other benefits.

What if *the computer*. . .

- . . . does not understand?
- . . . misunderstands?

Success criteria for today

Minimal

- You can *in your head* evaluate the code which consists of print statements and assignments
- You can (somehow) run Python
- You can (somehow) execute Python scripts

Optimal

- You can execute Python scripts from VSCode
- You have installed the course software
- You have successfully submitted the ungraded hand-in test project