

Introduction to Programming

Practical Class #2

João Fonseca

joao.fonseca@novasbe.pt



NOVA SCHOOL OF
BUSINESS & ECONOMICS

Agenda

- Recap from theoretical class
- Jupyter Notebook Intro
- Introduction to the command line
- Writing your 1st Script

Some basic concepts

- A **program** is a sequence of instructions to be performed by a computer
- An **interpreted language** is a high-level language run and executed by an interpreter (a program which converts the high-level language to machine code and then executing on the go). Scripts in these languages are run on a statement by statement basis.
- A **compiled language** is typically implemented using a compiler to convert the program into machine code
- Python is an interpreted language

Advantages of using Python

- As a **high-level language** Python is highly readable, thus easier to learn
- **Extensive libraries**
- Large **community support**
- Ability to write complex operations with less effort
- Improves the programmer's productivity
 - **Reduces programming time** at the cost of increased processing time (compared to programming languages like Java, C or C++)
- **Excellent debugging environment.**

Disadvantages of using Python

- **Performance is not the best...**

Source: Pereira et. al.

Energy Efficiency across Programming Languages - How Does Energy, Time, and Memory Relate?

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Data Types

- **Integer:** Whole numbers – 1, 56, 386, 29...
- **Float:** Fractional values (values have digits on both sides of a decimal point) – 1.4, 38.111, 2.25693713, 1.0, 4.0
- **Boolean:** Two possible values to represent logic – True, False
- **String:** Characters – 'hey there', '7', 'th1s | 1s_ paul\'s_string'
 - (The “\” within a string is used to input special characters, such as “ ’ ”)

Some basic Python functions

- `print()`
- `input()`
- Math operations: `1+1`, `2-1`, `3*3`, `3/4`
- Logic operations: `2==8/4`, `3>1`, `6<=10` ...
- `int()`, `str()`, `float()`
- `list()`, `len()`, `range()`
- `min()`, `max()`, `sum()`
- `type()`

Jupyter Notebooks

- Allows you to interactively study the materials
- Open Introduction to Programming's Github repo:
 - https://github.com/joaopfonseca/introduction_to_programming
- In there you will see plenty of notebooks covering the fundamentals of Python programming
- Open the file:
 - 00_introduction_to_jupyter_notebooks.ipynb

Command Line Basics

- For an experienced user, the command line represents a much faster and easier method to perform many tasks
- Steep learning curve (it may seem overwhelming at times, but don't worry, just ask for help if you're having troubles!)
- The command line can be used to automate processes by converting repetitive tasks into one command
- Used to run scripts (only way to run Python programs outside of an IDE)
- Bulk operations become much easier
- Many operations cannot be performed in a regular GUI (Graphical User Interface), thus making it a powerful tool for programming
- If using cloud computing, it is very likely that the only way to access it is via the terminal
- ... looks cool 😊

Know when to use one over the other!

- Use the provided Cheat Sheet!
- Exercise:
 1. Launch the terminal window
 2. Install two Shell packages: tree and git
- What are they for?
 - tree: print a directory's structure
 - git: manage Github repositories (we will go through this later)

Package Managers

- Two main python package managers exist:
 - conda
 - pip
- Package managers automate the installation, upgrading, configuration and uninstallation processes
- Other package managers:
 - Homebrew (mac OS package manager – not built in)
 - Advanced Package Tool: apt (Ubuntu's and other Linux distros' package manager)
 - ...

Command Line Basics

- Use the provided Cheat Sheet!
- Let's go through some basic commands: ls, cd, pwd, rm, pwd, touch, mkdir, sudo
- Exercise:
 1. Launch the terminal window
 2. Print your working directory (you will see the directory for your home folder)
 3. Navigate to your Desktop and create a new directory (i.e., a folder – name it ip_class_exercises)
 4. Create a Python script in your new directory: ip_class_exercises. Name it 01_my_first_script.py
 5. Check which files you have in that directory (**through the terminal!**)

Let's write our first Python script!

- Exercise:
 - Open Spyder
 - Open your newly created script: 01_my_first_script.py
 - Program it to print “Hello World!” and save the file
 - Run it in the IDE (reminder: IDE=Integrated Development Environment)
 - Once finished, close the IDE and run the script in the terminal

Let's write our first Python script!

- **Homework:** if statements!
- Exercise:
 1. Create a new python script. Name it 02_homework_calcs_and_ifs.py
 2. Create an input: “Choose one: [sum two values] or [birth date to month converter]”
 3. Create an if statement: if variable==“sum values” ask for two inputs, convert both to float and print their sum.
 4. Create another if statement: if variable==“birth date to month converter” ask for the year and month of birth and print the number of months lived until September 2018