

Programming for Aml

MOTIVATIONS AND GOALS

Why Aml needs programming? Define the goals and requirements of software development for an Ambient Intelligent system



POLITECNICO
DI TORINO



Ambient Intelligence systems: **digital environments** that **pro-actively**, but **sensibly**, support people in their daily lives.

How?

- By **blending systems and devices** in the environment
- By **adding software** to coordinate different components and make them behaving as a single organism
- By designing this organism to be **“interactive”**, **“supportive”** and **“sensible”**

Software

- Goal
 - coordinate the project components
 - make them “interactive”, “supportive” and “sensible”
- Requirements
 - focus on features
 - effectively tackle “intelligence” design
 - solve “real” problems
 - avoid / limit programming idiosyncrasies

Python

- Solve “real” problems
- Smooth learning
- Avoid focusing on mathematical abstraction, only
- Limit distraction from
 - Low-level syntax issues
 - Compilation
 - Counter-intuitive concepts

Python

AN OVERVIEW

A short overview of Python, including a bit of history, motivation for its adoption in the Ambient Intelligence course, and basic programming concepts



POLITECNICO
DI TORINO

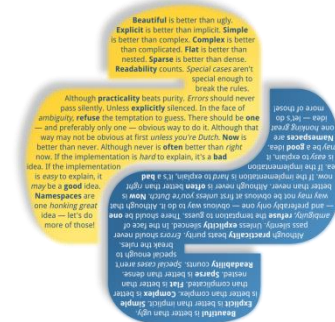


What is Python?

- An **easy to learn, powerful** programming language
- An **ideal language** for scripting and **rapid application development** in many areas on most platforms

Identikit

- First appeared in 1991
- Designed by Guido van Rossum
- General purpose
- High level
- Emphasis on code readability and conciseness
- Website
 - <http://www.python.org>
- We will use Python 3
 - not Python 2



About (programming) languages...

- High level vs. low level languages
- Interpreted vs. compiled languages

What is the difference?

High level languages

- Near to human-level abstraction
 - Short, expressive, easy to read
- Portable
 - Can be executed on different platform with few or none changes
- Must be translated into low-level code for actual execution

Hello, world! (high level)

```
#include <stdio.h>

int main()
{
    printf("Hello, world!\n");
    return 0;
}
```

Low level languages

- Directly executable
 - No translation needed
- Typically more efficient
 - They are designed for very specific hardware
- Platform dependent
 - Must be re-written for execution on different platforms
- Difficult to write (and read)
 - Near to the machine code

Hello, world! (low level)

```
.section      .rodata
string:
.ascii "Hello, world!\n"
length:
.quad . -string      #Dot = 'here'

.section      .text
.globl _start      #Make entry point visible to linker
_start:
movq $4, %rax      #4=write
movq $1, %rbx      #1=stdout
movq $string, %rcx
movq length, %rdx
int $0x80          #Call Operating System
movq %rax, %rbx    #Make program return syscall exit status
movq $1, %rax      #1=exit
int $0x80          #Call System Again
```

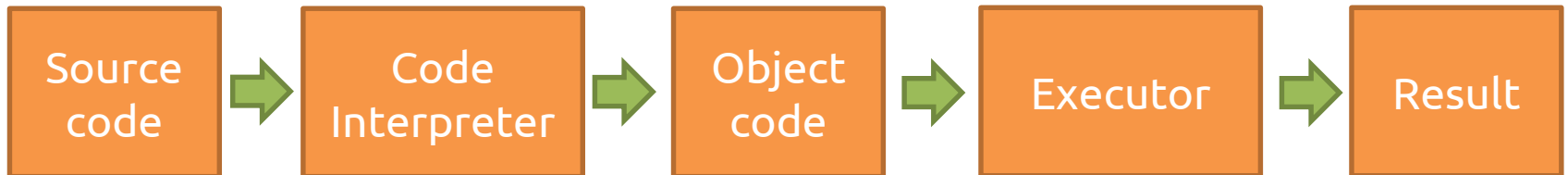
Interpreted languages

- Line by line translation and execution



Compiled languages

- Completely translated into low-level code before execution



Python is interpreted

- Interactive mode

- Type the program and the interpreter displays the result

```
>>> 1+1  
2
```

- Script mode

- Store the code in a file, and use the interpreter to execute the contents

```
python myscript.py
```


Getting started

PYTHON INSTALL



Python 3.x Availability

- High Level
 - Available for the major platforms
- Linux
 - Typically preinstalled
 - or available through package manager
 - check
 - type "python3" in a terminal
- Windows / macOS
 - Should be explicitly installed

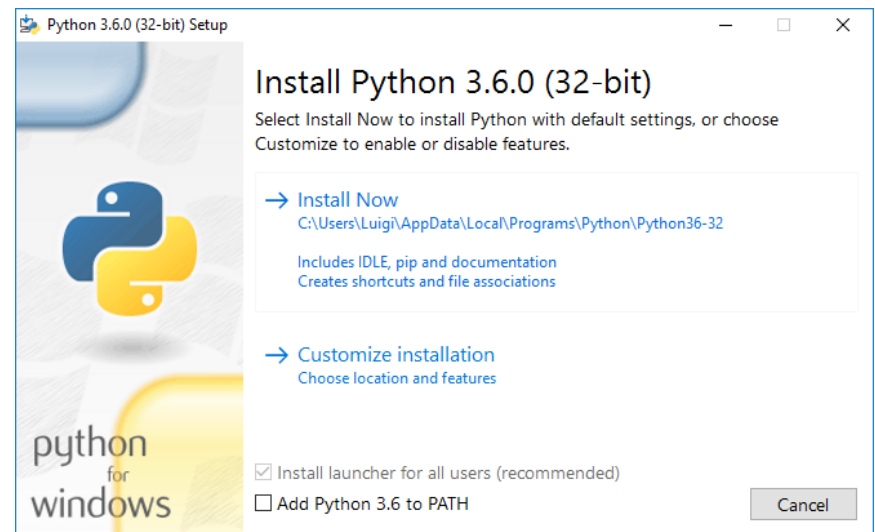
We will use



GNU / Linux

Windows Installation

- Check the latest 3.x version
 - <http://www.python.org>
- Download the .exe installer
 - follow the wizard throughout installation
- Open-up a terminal
 - Win(+R) > cmd
 - python --version



```
C:\>python --version
Python 3.6.0
```

macOS Installation

- Check the latest 3.x version
 - <http://www.python.org>
- Download the .pkg installer
 - follow the wizard throughout installation
- Alternatively, use homebrew
 - <https://brew.sh/>
 - `brew install python3`
- Open-up a terminal
 - spotlight > terminal
 - `python3 --version`



```
[luigi@gallifrey:~/ > python3 --version  
Python 3.6.0
```

Integrated Development Environment (IDE)

A **software application** that provides comprehensive facilities to computer programmers for software development.

An IDE normally consists of a **source code editor**, **build** automation tools and a **debugger**.

Most modern IDEs offer Intelligent code completion features.

Python IDE

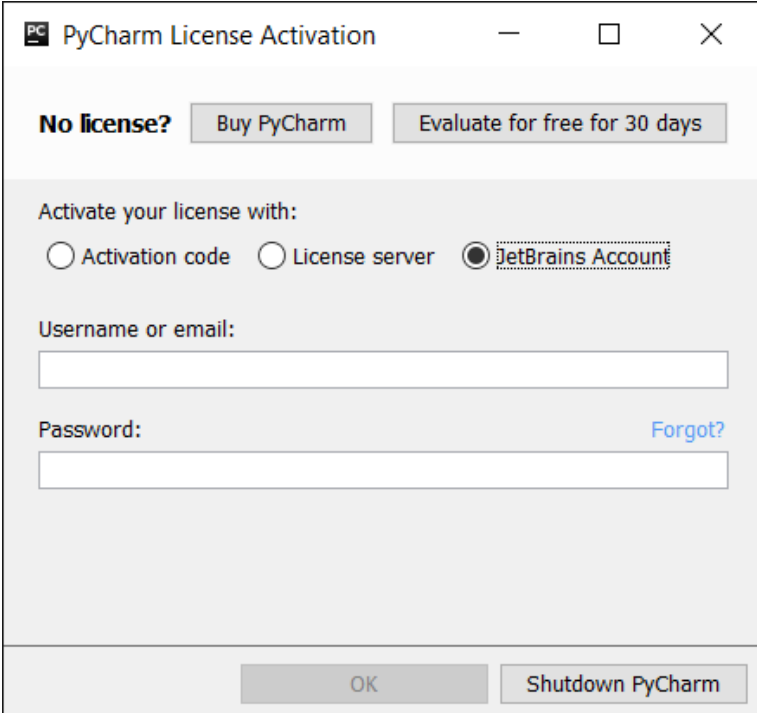
- Some choices available
- We use JetBrains PyCharm
 - Professional Edition
 - <https://www.jetbrains.com/pycharm/>
- PyCharm is a **commercial** product
- JetBrains provide a **free** license for students
 - <https://www.jetbrains.com/student/>
 - apply with your @studenti.polito.it e-mail address!

PyCharm Installation

- Apply for a free JetBrains license
 - <https://www.jetbrains.com/student/>
- Download PyCharm Professional Edition
 - <https://www.jetbrains.com/pycharm/download>
 - available for Windows, Linux and Mac
- On Windows / Mac
 - double click on the downloaded file
- On Linux
 - extract the .tar.gz file where you want to install the IDE

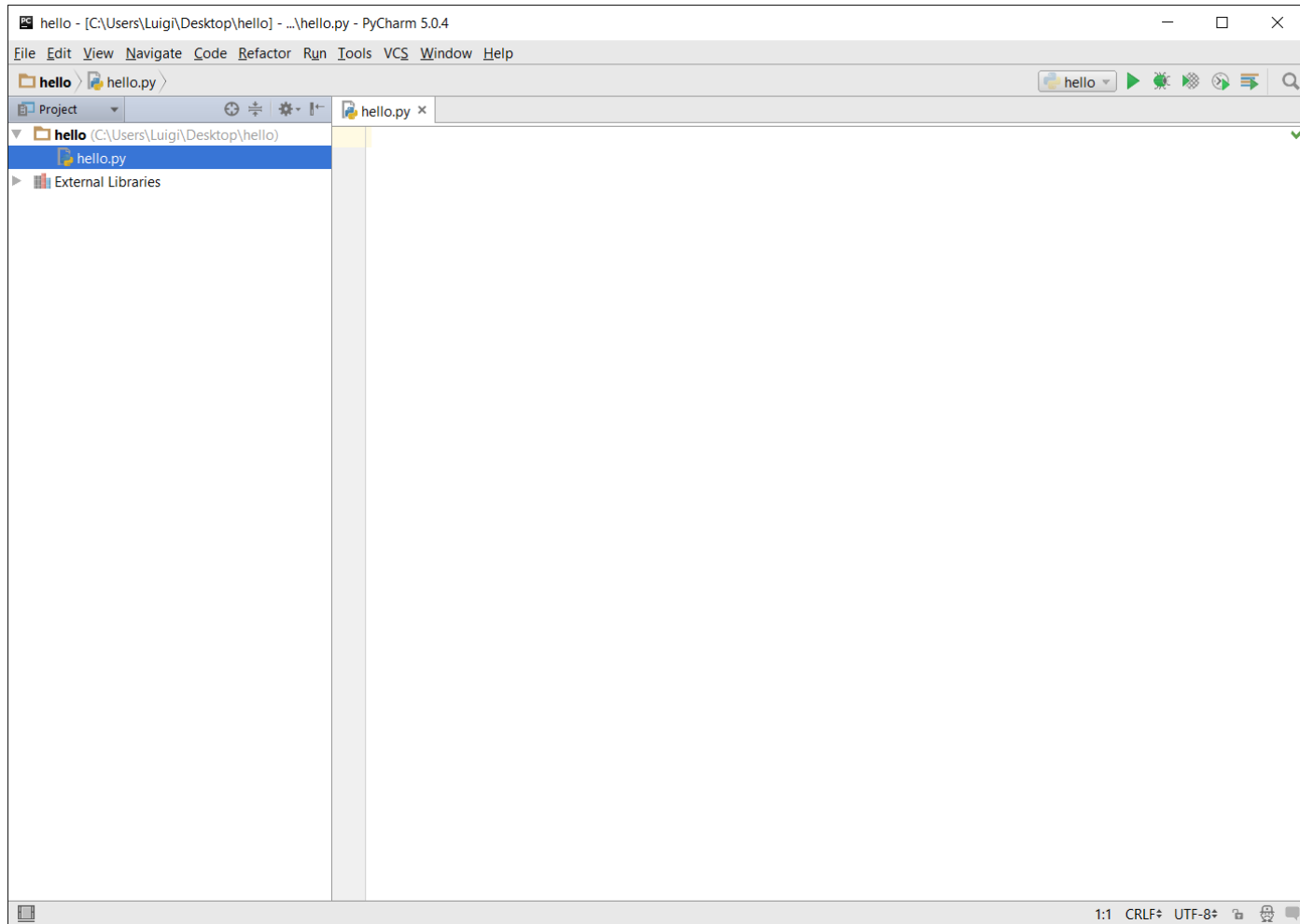
PyCharm Installation

- Open PyCharm
- Insert your JetBrains credentials



The image shows a screenshot of the "PyCharm License Activation" dialog box. The window title is "PyCharm License Activation". At the top, there is a "No license?" section with two buttons: "Buy PyCharm" and "Evaluate for free for 30 days". Below this, the text "Activate your license with:" is followed by three radio button options: "Activation code", "License server", and "JetBrains Account". The "JetBrains Account" option is selected. Underneath, there are two input fields: "Username or email:" and "Password:". A "Forgot?" link is located to the right of the password field. At the bottom of the dialog, there are two buttons: "OK" and "Shutdown PyCharm".

Hello, world!

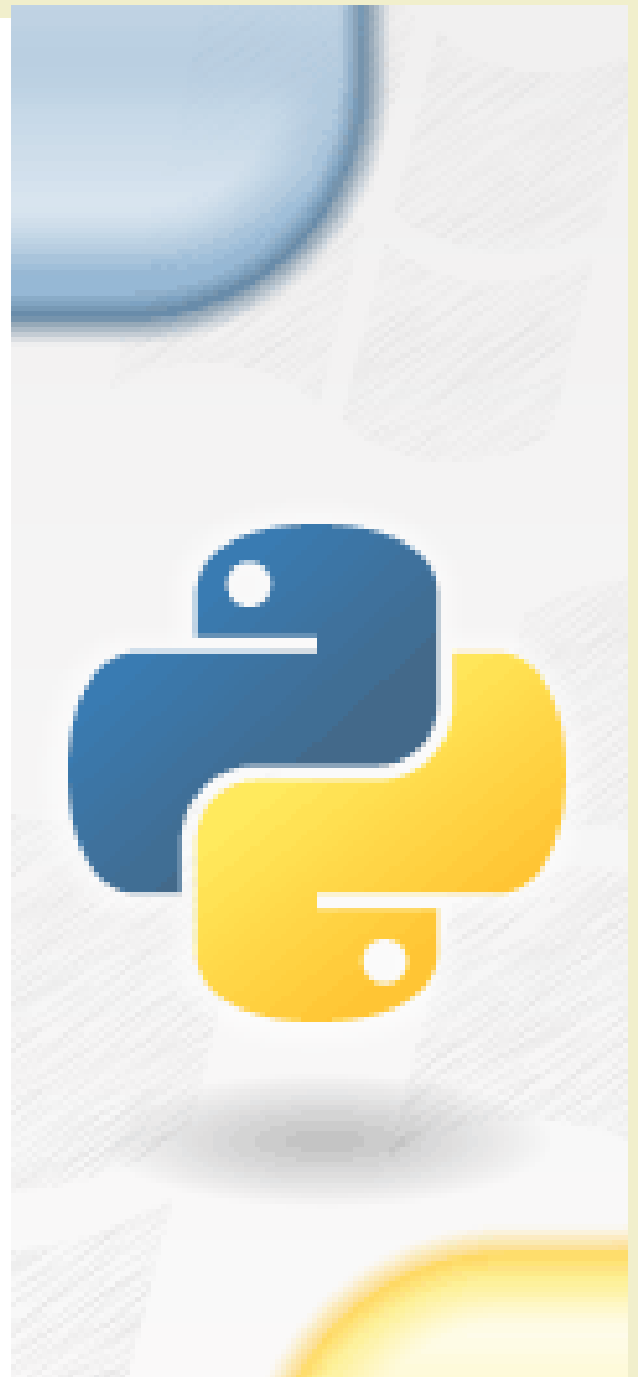


Questions?




01QZP AMBIENT INTELLIGENCE

Luigi De Russis

luigi.derussis@polito.it



License

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 4.0)” License.
- You are free:
 - to **Share** - to copy, distribute and transmit the work
 - to **Remix** - to adapt the work
- Under the following conditions:
 - **Attribution** - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). 
 - **Noncommercial** - You may not use this work for commercial purposes. 
 - **Share Alike** - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. 
- To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>