

Programming for Aml

MOTIVATIONS AND GOALS

Why Aml needs programming? Defining the goals and requirements of software development for Aml



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»
- Functional Requirements
 - Focus on features
 - Effectively tackle «intelligence» design
 - Solve «real» problems
 - Avoid / Limit programming idiosyncrasies

Solution

- 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, justification for its adoption in the Ambient Intelligence: Technology and Design course, and basic programming concepts



What is Python?

- Python is an **easy to learn, powerful** programming language.
- It is an **ideal language** for scripting and **rapid application development** in many areas on most platforms.

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)

`print` 'Hello world'

```
>>> print 'Hello World!'
Hello World!
```

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

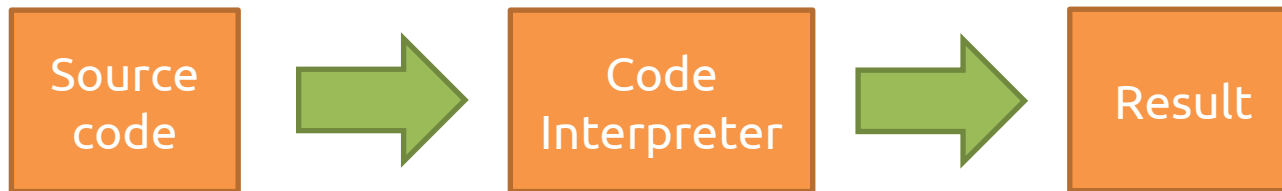
Hello world (low level)

```
.section      .rodata
string:
.asciz "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

- Two ways to use the interpreter:
- 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

3/9/2015

Programming for Aml



Linux vs Windows

- High Level
 - Available for both platforms
- Linux / Unix
 - Typically pre-installed
 - Already used for several tasks
- Windows
 - Should be explicitly installed
- Mac
 - Typically pre-installed

We will use



GNU / Linux

Windows Installation

- Check the latest 2.7.x version
 - <http://www.python.org>
- Download the .msi installer
 - follow the wizard throughout installation
- Open-up a terminal
 - Win(+R) > cmd
 - python --version



```
bonino@CALIPSO ~  
$ python --version  
Python 2.7.6  
bonino@CALIPSO ~  
$
```

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 PyDev
 - <http://pydev.org/>
- Installed as an extension of the Eclipse IDE
 - <http://www.eclipse.org/>

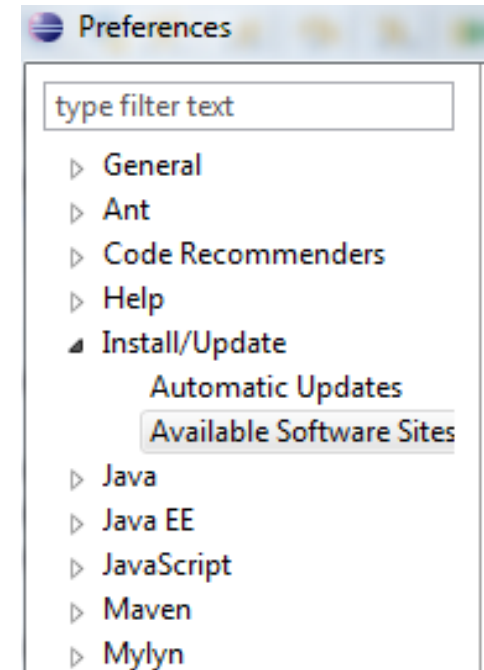


PyDev Installation

- Download Eclipse 4.3.2 standard
 - <https://www.eclipse.org/downloads/>
 - Check your os version (Linux/Windows/Mac, 32/64bit)
 - Unzip the eclipse file where you want-it to be installed
 - `tar -xvfz eclipse-standard-kepler-SR2-linux-gtk-x86_64.tar.gz`
(on Linux)
 - `eclipse-standard-kepler-SR2-win32.zip`

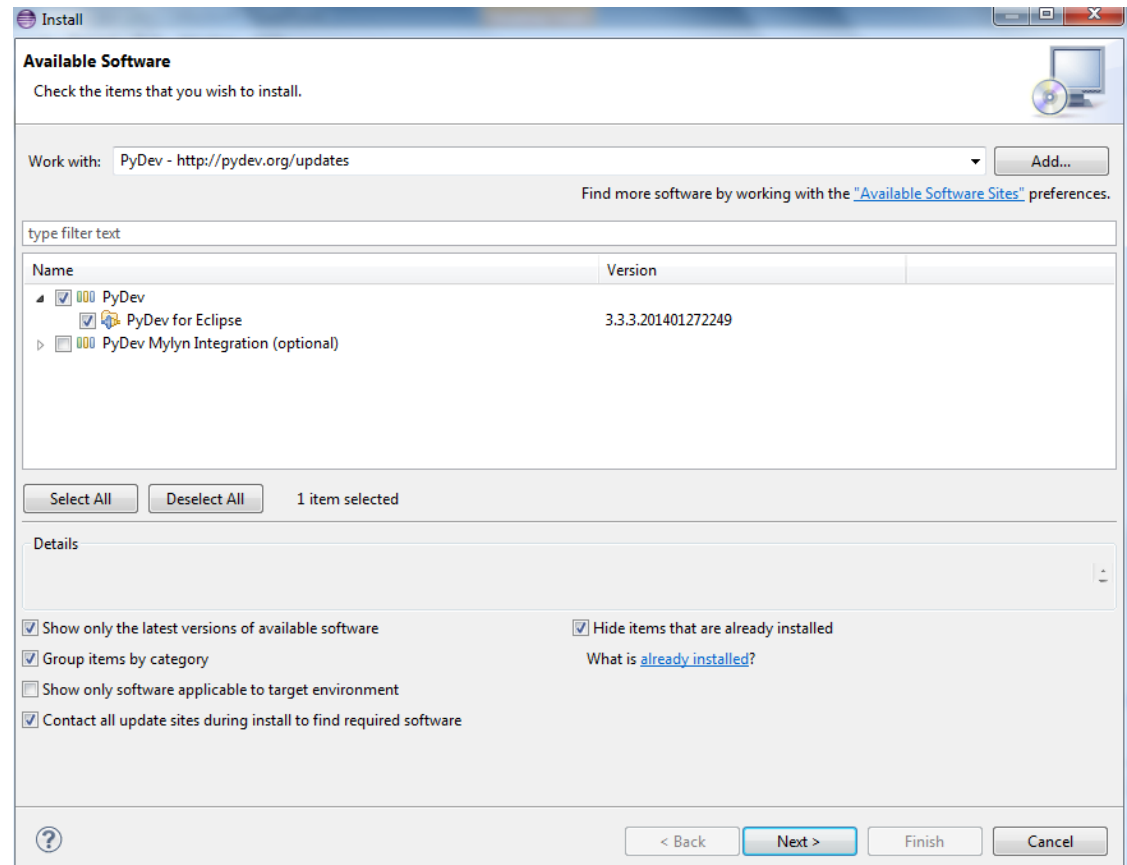
PyDev Installation 2

- Open the eclipse preferences
 - Window > Preferences > Install/Update > Available Software Sites
- Add the PyDev update site
 - <http://pydev.org/updates>
- Install the new software
 - Help > Install New Software...

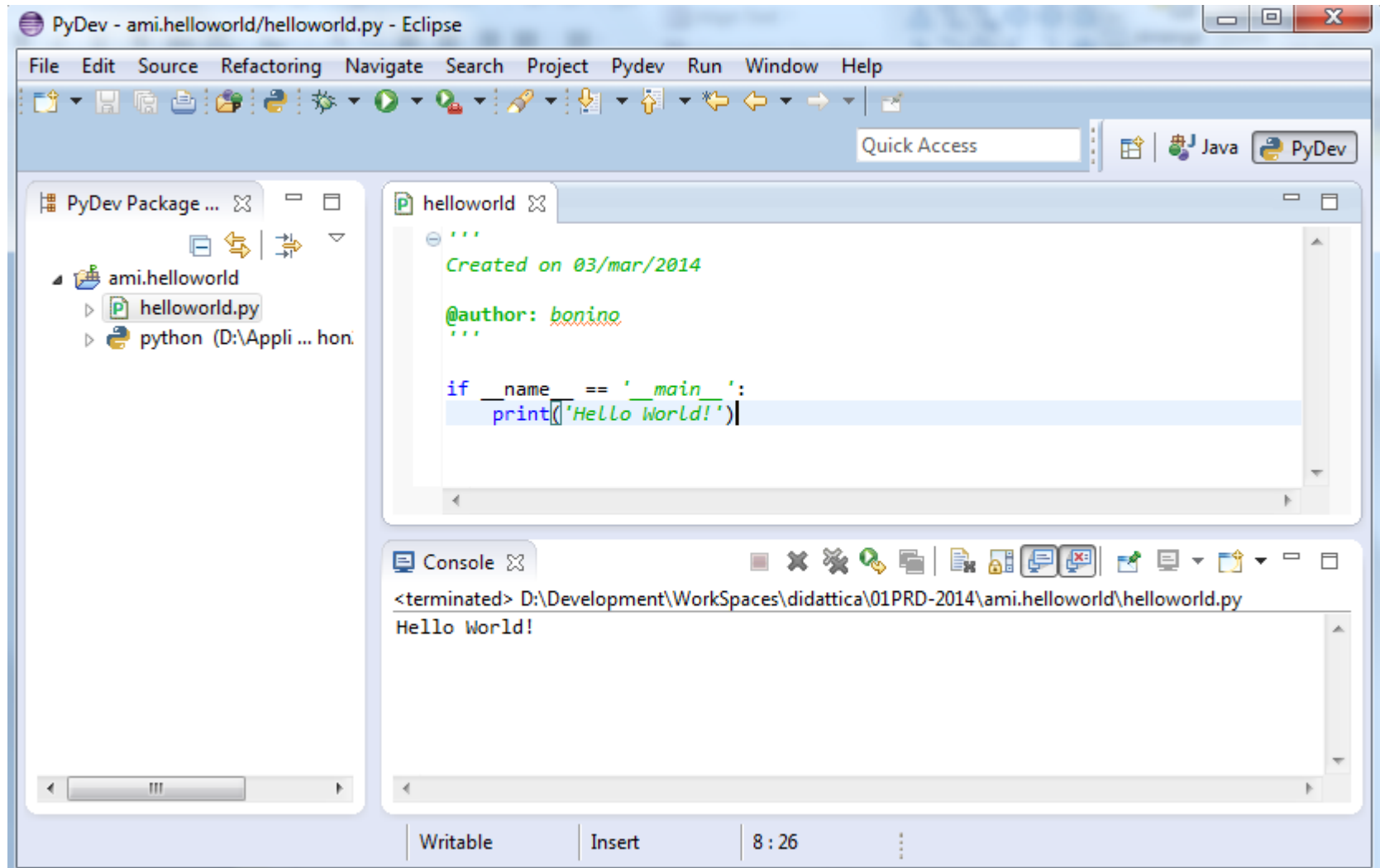


PyDev Installation 3

- Select the just created update site
- Select PyDev for Eclipse



Hello world!



Hands on

Define a script for saying «Hello!» to the whole Aml course. Do it using the interactive interpreter and then by writing a script file

```
print ('Hello Aml 2014!')
```

```
if __name__ == '__main__':  
    print('Hello World!')
```

Hands on

Define a script for computing the sum of 2 numbers. Do it using the interactive interpreter and then by writing a script file

```
>>>13+22
```

```
if __name__ == '__main__':  
    print(13+22)
```

What is a program?

A program is a sequence of instructions that specifies how to perform a computation.

- Basic instructions
 - **input:**
 - Get data from the keyboard, a file, or some other device.
 - **output:**
 - Display data on the screen or send data to a file or other device.
 - **math:**
 - Perform basic mathematical operations like addition and multiplication.
 - **conditional execution:**
 - Check for certain conditions and execute the appropriate code.
 - **repetition:**
 - Perform some action repeatedly, usually with some variation.

Programming

the process of breaking a large, complex task into smaller and smaller subtasks until the subtasks are simple enough to be performed with one of these basic instructions.

- error-prone
 - **Syntax error**
 - An error in a program that makes it impossible to parse (and therefore impossible to interpret).
 - **Semantic error**
 - An error in a program that makes it do something other than what the programmer intended
 - **Runtime Error**
 - An error that is detected while the program is running.

Programming 2

- **bug:**
 - An error in a program.
- **debugging:**
 - The process of finding and removing any of the three kinds of programming errors.

Questions?

01PRD AMBIENT INTELLIGENCE: TECHNOLOGY AND DESIGN

Dario Bonino
bonino@ismb.it

