Introducción al desarrollo de software

Publicación 1.0

Emiliano López

13 de May de 2015

Índice general

Contents:

CAPÍTULO 1

Introducción

Docente: Emiliano López (emiliano [dot] lopez [at] gmail [dot] com)

En el presente capítulo introduciremos los conceptos necesarios para desarrollar los primeros algoritmos computacionales. Además, se explican las herramientas necesarias para llevar a cabo el desarrollo y sus diferentes alternativas.

1.1 Instalando Python

Actualmente existen dos versiones de Python comúnmente utilizadas, la versión 2 y 3, ambas son completamente funcionales y muy utilizadas. En este curso nos basaremos en la versión 3.

1.1.1 Windows

Para instalar Python en una máquina con Windows, debemos seguir los siguientes pasos:

- Apuntar el navegador a: https://www.python.org/downloads/windows/
- Ir al link de la última versión disponible (por ej: latest python 3 relase)
- En la sección Files, descargar el instalador correspondiente a su arquitectura (64/32 bits), por ej: https://www.python.org/ftp/python/3.4.3/python-3.4.3.msi
- Ejecutar el instalador (por ej: python-3.4.3.msi) aceptando las opciones por defecto

1.1.2 GNU/Linux

En los sistemas operativos serios, es muy probable que ya contemos con el intérprete instalado, incluso en sus dos versiones. Para instalarlo utilizando los administradores de paquetes debemos ejecutar los siguientes comandos desde una terminal:

Para sistemas basados en Debian (como Ubuntu o sus derivados):

```
apt-get install python3
```

1.2 Entornos de programación

1.2.1 El intérprete interactivo

Ya con el intérprete de Python podemos comenzar programar. Si ejecutamos en una terminal `python3`, ingresaremos al intérprete en modo interactivo y veremos una salida similar a la siguiente:

```
Python 3.4.2 (default, Oct  8 2014, 10:45:20)
[GCC 4.9.1] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Podemos ejecutar algunas operaciones matemáticas para corroborar su funcionamiento.

```
>>> 2*5
10
>>> 2*5+10
20
>>> -3*19+3.1415
-53.8585
>>>
```

1.2.2 El intérprete interactivo mejorado

IPython[1] es una interfaz mejorada del intérprete nativo. Se lo puede utilizar en modo consola o a través de una interfaz web. La instalación en GNU/Linux es: `apt-get install ipython3`.

La ejecución de ipython desde una terminal nos arroja una pantalla similar a la siguiente:

```
emiliano@pynandi:~ $ ipython3
Python 3.4.2 (default, Oct  8 2014, 10:45:20)
Type "copyright", "credits" or "license" for more information.

IPython 2.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

Otra alternativa muy interesante son los notebooks de ipython, una interfaz que permite programar utilizando el navegador web como entorno. No entraremos en detalle ya que posteriormente analizaremos su funcionamiento. Se debe ejecutar en una terminal `ipython3 notebook` y esto abrirá el navegador por defecto con el entorno cargado.

1.3 Entorno integrado de desarrollo (IDE)

Un IDE es un entorno que nos facilita las tareas a la hora de programar. Consiste en la integración de un editor de texto, con características de resaltado de sintaxis autocompletado -entre otras-, y el intérprete de Python. Existen cientos de entornos muy buenos, como por ejemplo Spyder[2], PyCharm[3] o Ninja-IDE[4]. Para el presente curso, nos basaremos en

[1] http://ipython.org
[2] https://github.com/spyder-ide/spyder
[3] https://www.jetbrains.com/pycharm
[4] http://ninja-ide.org

Ninja-IDE, software libre que ha sido desarrollado por la comunidad de Python Argentina, PyAr[5].

Una lista bastante completa sobre las IDEs disponibles pueden encontrarse en la wiki oficial de Python[6]

1.4 El primer programa "Adiós mundo!"

El acercamiento inicial a un lenguaje de programación suele ser con el archiconocido programa "Hola mundo". Consiste simmplemente en un programa que muestra en pantalla ese mensaje.

Renunciando a cualquier pretensión de originalidad comenzaremos del mismo modo, pero despidiéndonos. Para esto utilizaremos la instrucción *print()* pasando el mensaje de despedida entre comillas, a continuación la instrucción y su salida correspondiente.

```
print("Adios mundo cruel!")
```

Adios mundo cruel!

Ahora bien, es muchísimo más lo que podemos hacer programando además de saludar cordialmente. Veamos los elementos de un programa que nos permitirán realizar tareas más complejas y entretenidas.

1.5 Elementos de un programa

1.5.1 Variables

1.5.2 Operadores relacionales y lógicos

1.5.3 Entrada y salida de datos

1.6 Ejercicios

[5] http://python.org.ar
[6] https://wiki.python.org/moin/IntegratedDevelopmentEnvironments

CAPÍTULO 2

Estructuras de control

2.1 Condicionales

2.1.1 if

2.1.2 else

2.1.3 Estructuras anidadas

2.1.4 elif

2.2 Repeticiones

2.2.1 while

2.3 Ejercicios

```
for i in range(1,10):
    if i%2==0:
        print(i,"es par")
    else:
        print(i, "es impar")
```

```
1 es impar
2 es par
3 es impar
4 es par
5 es impar
6 es par
7 es impar
8 es par
9 es impar
```

CAPÍTULO 3

Más estructuras de datos y control

3.1 Listas

3.2 for

3.3 Manipulando textos con strings

CAPÍTULO 4

Funciones, archivos, diccionarios

4.1 Definiendo funciones

4.1.1 Variables globales y locales

Agrupando el código en módulos

4.2 Números aleatorios

4.3 Lectura y escritura de archivos

4.4 Diccionarios

CAPÍTULO 5

Clases y objetos

CAPÍTULO 6

This is a Title

That has a paragraph about a main subject and is set when the '=' is at least the same length of the title itself.

6.1 Subject Subtitle

Subtitles are set with '-' and are required to have the same length of the subtitle itself, just like titles.

Lists can be unnumbered like:

- Item Foo
- Item Bar

Or automatically numbered:

1. Item 1
2. Item 2

6.2 Inline Markup

Words can have *emphasis in italics* or be **bold** and you can define code samples with back quotes, like when you talk about a command: `sudo` gives you super user powers!

CAPÍTULO 7

Indices and tables

- *genindex*
- *modindex*
- *search*