

1. Introducció

En aquest capítol presentem el llenguatge de programació Python i aprendrem a instal·lar-lo al nostre ordinador. També aprendrem a executar programes en Python. Per últim presentem el Jupyter Notebook, una aplicació web que permet crear i visualitzar documents que incorporen tant text com programes que es poden executar dins del mateix document.

D'aquest capítol disposes dels següents arxius:

- El capítol en PDF: 01-Introduccio-cat.pdf
- El programa de prova: [prova1.py](#)
- L'arxiu de Jupyter: [cap1-cat.ipynb](#)
- Enllaç al notebook a Google Colab: <https://colab.research.google.com/github/aoliverg/python/blob/master/notebooks/cap1-cat.ipynb>

1.2. Sobre les diferents versions de Python

Actualment conviuen dues versions principals de Python, la 2 i la 3. Aquestes dues versions no són compatibles entre si ja que hi ha una sèrie de canvis importants. En un mateix ordinador poden conviure les dues versions sense problemes. La versió 2 encara es manté ja que hi ha una gran quantitat de llibreries que encara no s'han portat a la versió 3. Si tens les dues versions de Python instal·lades a l'ordinador haureu de tenir molt clar com s'executa una i l'altra. Aquest detall el veurem en els subapartats 1.4 i 1.5.

En tot aquest llibre farem servir la versió 3 de Python, ja que és la més moderna i la que en un període de temps no gaire llarg substituirà completament a la versió 2. A mes, el Natural Language Toolkit (que veurem al capítol 4) ja està disponible per la versió 3 de Python.

1.3. Instal·lació de Python

Abans d'instal·lar Python hem de verificar si ja el tenim instal·lat al nostre sistema. Si treballem amb Windows és molt poc probable que tingueu instal·lat l'interpret de Python (si no és que tu mateix o algun altre usuari l'hagi instal·lat anteriorment). En canvi, si treballem amb Linux o Mac és pràcticament segur que tindreu algun interpret de Python instal·lat. Caldrà verificar si és de la versió 2, de la 3 o bé de totes dues versions alhora.

Per verificar si teniu instal·lat Python entreu a un Terminal (en Mac o Linux) o a una pantalla de Símbol de sistema (en Windows) i escriviu:

```
python  
i pitjeu Enter.
```

Ens podem trobar en les següents situacions:

1. Si no tenim Python instal·lat apareixerà un missatge de l'estil: Python no es reconeix com a un nom de programa (o missatge similar en Windows) o Python: command not found (o missatge similar en Linux o Mac).
2. Si tenim l'interpret de Python instal·lat i és d'una versió 2 entrarem en l'interpret intereactiu (mireu secció 1.4. L'interpret interactiu) i apareixerà un missatge com el següent:

```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)  
[GCC 4.8.2] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
(per sortir de l'interpret interactiu escriu exit() i Enter)
```

Fixeu-vos que en la primera línia s'indica la versió de Python (en l'exemple la 2.7.6). Com que es tracta d'una versió 2, no ens servirà per executar els programes d'aquest llibre. Que tinguem un interpret de la versió 2 no

1. Introducció

significa que no tinguem un de la versió 3 instal·lat. Per verificar-ho escriurem al Terminal o pantalla de símbol de sistema:

```
python3
```

Ara, si apareix un missatge Python no es reconeix com a un nom de programa (o missatge similar en Windows) o Python: command not found (o missatge similar en Linux o Mac) voldrà dir que ni tenim el Python versió 3 instal·lat al sistema. Si en canvi apareix:

```
Python 3.4.3 (default, Oct 14 2015, 20:28:29)
```

```
[GCC 4.8.4] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```

veiem que tenim tant la versió 3 com la 2 instal·lades en el sistema. Llavors, cal recordar que per executar l'interpret interactiu de la versió 3 (que és el que necessitem per als programes d'aquest llibre) haurem d'escriure `python3`.

3. Si només tenim la versió 3 de Python, quan escrivim `python` i enter ens apareixerà directament l'interpret interactiu de la versió 3 i per executar-lo només caldrà escriure `python`.

Llavors, si no disposem d'un interpret de la versió 3 l'haurem d'instal·lar seguint les següents instruccions:

La plana web des d'on podem trobar tota la informació necessària sobre el llenguatge de programació Python, fer les descàrregues de l'interpret, etc. és www.python.org. Un cop en aquesta plana anem a Download (o directament a l'enllaç www.python.org/downloads/). El primer que ens demanarà és que triem la versió 3 o 2. Triarem la versió 3 més nova que aparegui. Python s'instal·la com qualsevol altre programa. El que és molt important tenir en compte és que en una de les primeres pantalles de l'assistent d'instal·lació és imprescindible marcar l'opció Add Python 3.X to Path, com es mostra a la figura següent:



Per verificar si funciona escriviu en una pantalla de Símbol de sistema o Terminal `python` o `python3` (tal i com s'explica una mica més a dalt en aquesta mateixa secció).

1.4. L'interpret interactiu

Hi ha dues maneres principal d'executar un programa en Python. Podem escriure línies de codi en l'interpret interactiu i obtenir una resposta cada línia (per aquest motiu a aquest tipus d'execució se li anomena sessió interactiva) o bé podem escriure el codi d'un programa en un arxiu de text i executar-lo (com s'explica en la secció següent).

L'interpret interactiu serà de gran utilitat per a provar instruccions o petits programes abans d'escriure'ls en un editor de text. Fixeu-vos en el següent exemple com l'interpret va executant les instruccions que li indiquem.

```
Python 3.4.3 (default, Oct 14 2015, 20:28:29)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hola")
Hola
>>> a=2
>>> b=3
>>> print(a+b)
5
```

A més, l'interpret interactiu ens podrà donar informació sobre instruccions, funcions, classes i tipus de variables. Per exemple, si escrivim `help("print")` en l'interpret interactiu, ens apareixerà la següent informació:

Help on built-in function print in module builtins:

```
print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.

(END)
```

Aquesta pot ser una manera ràpida i sempre disponible d'accedir a la documentació de Python.

1.5. Execució de programes en Python

L'ús de l'interpret interactiu està recomanat únicament per a provar instruccions o petits fragments de codi o per consultar la informació sobre les instruccions, funcions, classes i tipus de variable. Quan fem programes més complexos és molt millor escriure'ls en un fitxer de text fent servir un bon editor de textos (mireu la secció següent). Per exemple, si el nostre programa es diu [prova1.py](#) (tot i que no és imprescindible, és recomanable que els programes en Python duguin l'extensió `.py`), per executar aquest programa només cal obrir una pantalla de Símbol de sistema (o terminal) i escriure

```
python prova1.py
o
python3 prova1.py
```

(depenent de quantes versions de Python tingueu instal·lats al sistema, tal i com s'explica en la secció 1.2. En la resta d'aquest llibre posarem sempre `python` en els exemples).

És important tenir en compte que el programa s'ha de trobar al directori on ens trobem en el moment d'executar-lo. Si no és així tenim dues opcions:

- Anar al directori corresponent fent servir les instruccions adequades d'MS-DOS (en Windows) o d'UNIX (en Linux i Mac).
- Donar la ruta completa d'on es troba el programa. Per exemple, si el programa es troba a la carpeta exemples de la unitat C: haurem d'escriure:

```
python C:\exemples\prova1.py
```

Si treballem en Linux o mac i el nom d'usuari es Maria i el directori exemples haurem d'escriure

```
python /home/Maria/exemples/prova1.py
```

En aquest punt cal recordar que és possible fer servir totes les opcions del sistema operatiu per indicar rutes senceres d'altres maneres (com per exemples de manera relativa al directori on ens trobem).

```
print("Hola")  
  
a=2  
b=3  
  
print(a+b)
```

1.6. Ús d'un bon editor de textos

Per escriure els nostres programes farem servir editors de textos. Convé no confondre un editor de textos amb un processador de textos. Un editor de textos és un programa que permet crear i editar arxius de text. En canvi, un processador de textos permet crear documents amb diferents tipus de lletra i amb funcions com la creació de taules, inserció d'imatges, etc. Exemples de processadors de text són el Word de Microsoft i el Libre/Open Office. Cal evitar l'ús d'editors de text massa senzill, i especialment no fer servir en cap cas el Notepad de Windows per escriure els nostres programes en Python. Els següents editors de programari lliure són probablement els més recomanables:

- Per a Windows:
 - Notepad++ (<https://notepad-plus-plus.org/>)
 - Geany (<https://www.geany.org>)
- Per a Linux:
 - Geany (<https://www.geany.org>)
- Per a Mac:
 - L'editor que ve per defecte, l'Edit, és una bona opció.
 - Geany (<https://www.geany.org>)

1.7. Jupyter Notebook

El Jupyter Notebook és una aplicació web que permet crear i compartir documents que contenen tant text (en un format molt semblant al Wiki) i codi de programació (que en el nostre cas serà Python). Amb aquesta aplicació web el codi inserit al document es pot executar directament i observar els resultats.

Tots els capítols d'aquest llibre estaran acompanyats d'un document de Jupyter que repassaran el contingut i permetrà executar els programes directament. Un aspecte també molt interessant és el fet que aquest arxius són editables de manera que es poden fer petites modificacions del codi o ampliar la informació del text.

La instal·lació de Jupyter és senzill ja que només cal fer:

```
pip install jupyter
```

en una pantalla de Símbol de sistema o Terminal. Si tens dos versions de Python instal·lades i per cridar a Python 3 has de fer servir la instrucció python3, llavors per instal·lar paquets i llibreries per Python3 hauràs d'escriure pip3 enlloc de pip. Per poder instal·lar aplicacions cal accedir a aquesta comanda com a administrador. En Windows, busqueu el programa cmd i abans d'executar-lo seleccioneu-lo amb el ratolí i feu clic al botó dret, apareixerà un menú i seleccioneu Executa'l com a administrador. En Linux i Mac escriviu sudo abans de la instrucció.

Per executar Jupyter obrint un arxiu :

```
jupyter notebook cap1.ipynb
```

1.8. Google Colab

Google Colaboratory és un entorn gratuït que permet l'execució de Jupyter Notebooks directament des d'un navegador d'Internet. D'aquesta manera disposarem d'un entorn d'execució dels notebooks d'aquest llibre sense necessitat d'instal·lar res a l'ordinador. Es pot accedir a Google Colaboratory des de l'enllaç <https://colab.research.google.com/>.

Per executar directament els notebooks d'aquest llibre podeu fer servir l'enllaç que oferim al principi de cada capítol. Per exemple, per accedir al notebook corresponent a aquest capítol, podeu fer clic a l'enllaç:

<https://colab.research.google.com/github/aoliverg/python/blob/master/notebooks/cap1-cat.ipynb>

És possible que aparegui un missatge com el següent:

Warning: This notebook was not authored by Google.

This notebook is being loaded from [GitHub](#). It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook.

CANCEL

RUN ANYWAY

En aquest cas només caldrà fer clic al botó RUN ANYWAY.