

01 – Introducción al lenguaje de programación Python 3



Diego Andrés Alvarez Marín
Profesor Asociado

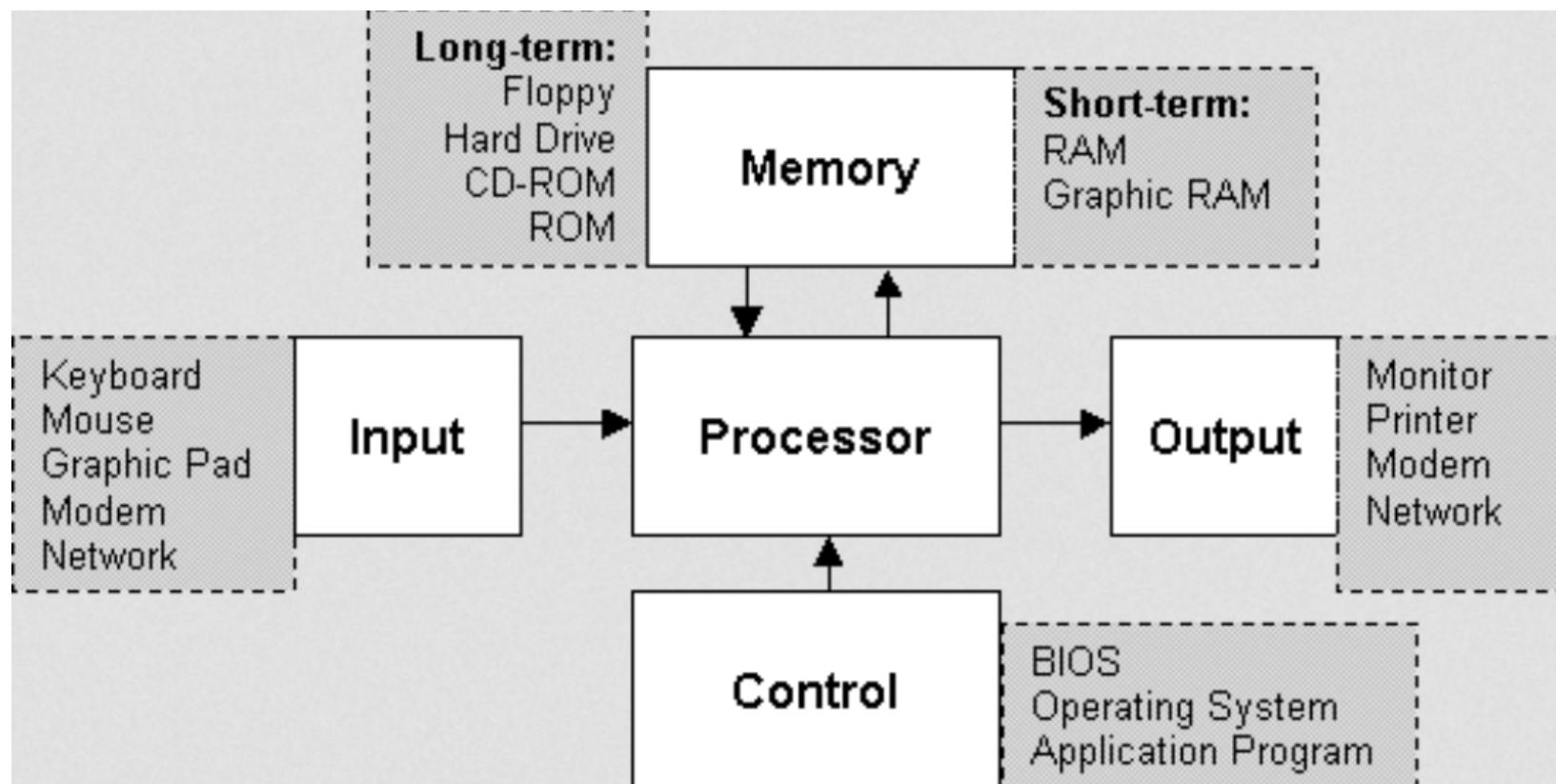
Universidad Nacional de Colombia
Sede Manizales

Tabla de contenido

- Arquitectura de un computador
- Sistemas operativos
- Bytes
- Lenguajes de programación (lenguaje de máquina, bajo, medio, alto nivel)
- Python
- Compilador vs Interprete

Arquitectura de un computador

- Procesador (CPU, memoria caché)
- Buses de datos (son las flechas en el dibujo)
- Memoria (de corto y largo plazo)
- Periféricos (entrada, salida)



Sistemas operativos

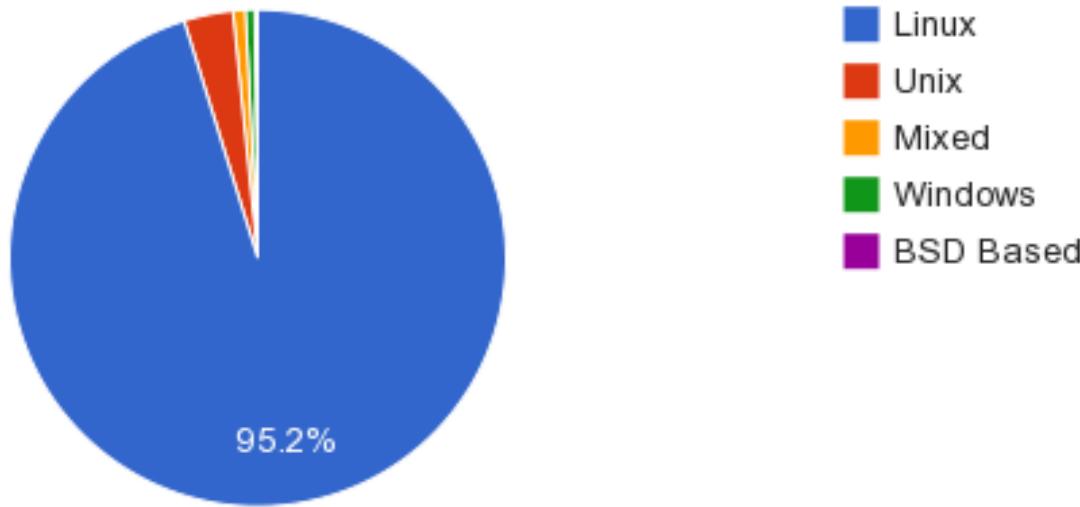
Un sistema operativo es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación.

- Microsoft Windows
- GNU/Linux (y su hijo Android)
- Mac OS X
- iOS
- Unix
- Solarix
- FreeBSD
- BlackBerry OS



Estadísticas a Junio 2013 con respecto al uso de sistemas operativos en los 500 computadores más poderosos del planeta

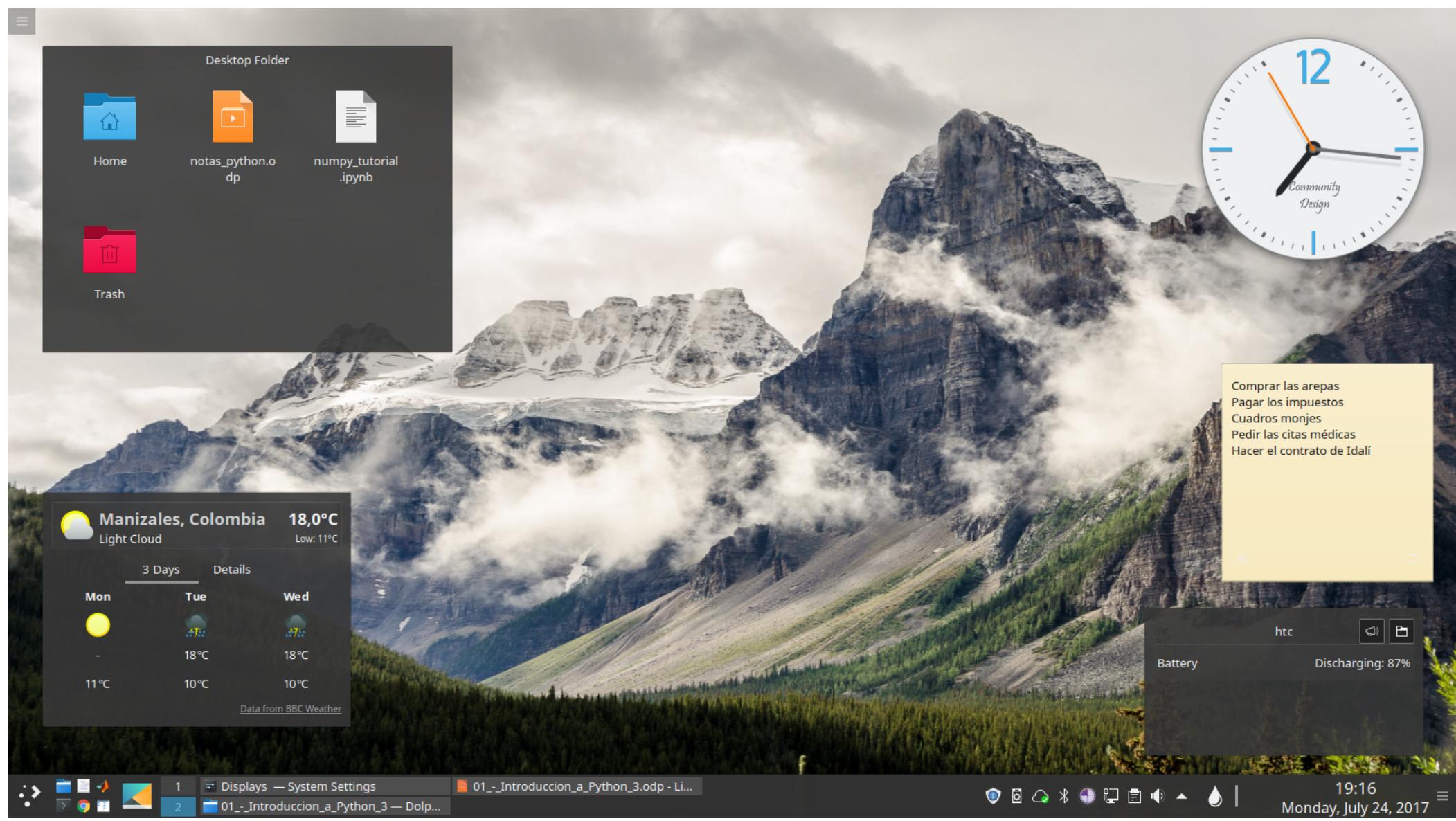
Operating system Family System Share



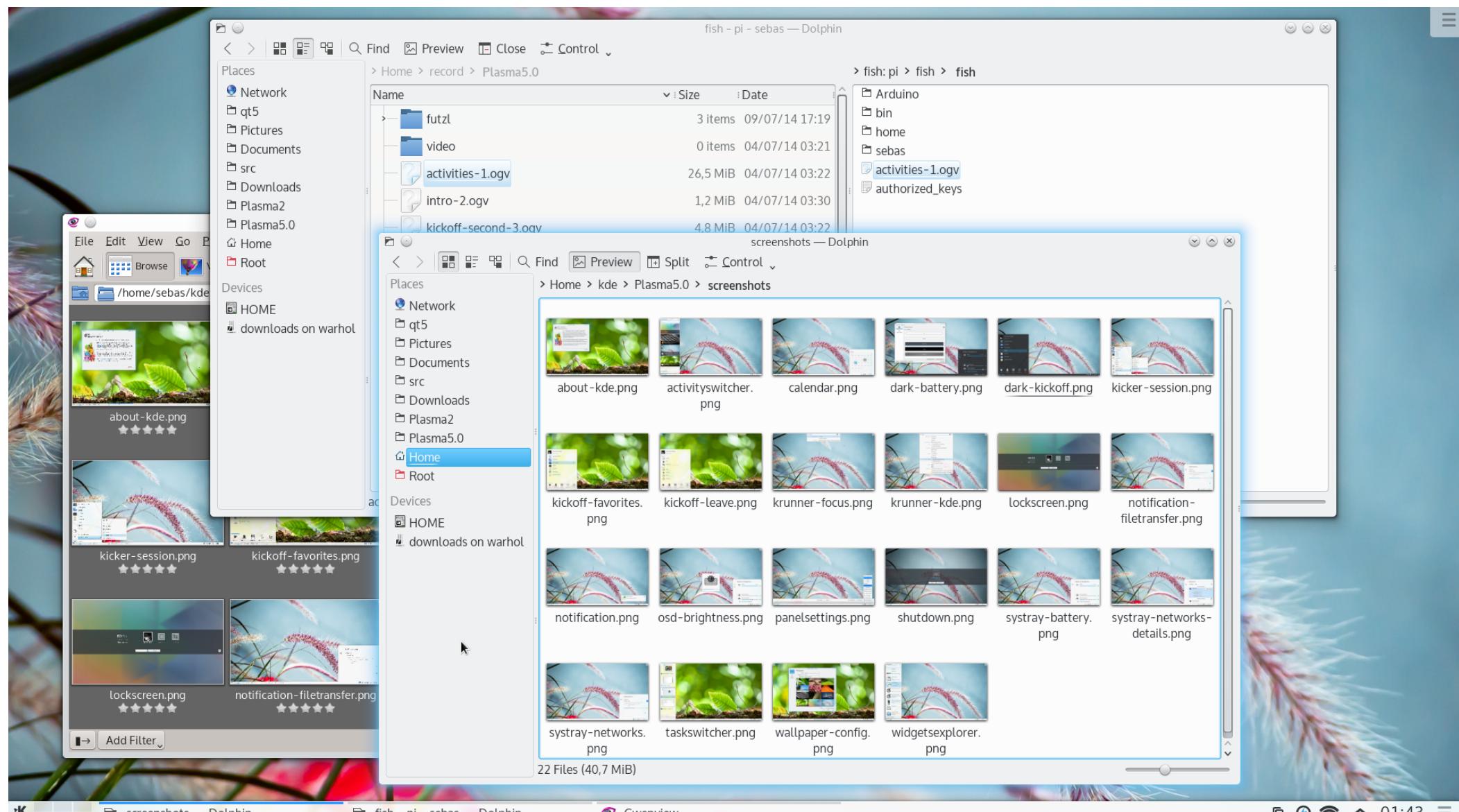
| Operating system Family | Count | System Share (%) | Rmax (GFlops) | Rpeak (GFlops) | Cores |
|-------------------------|-------|------------------|---------------|----------------|------------|
| Linux | 476 | 95.2 | 217,932,444 | 318,748,391 | 18,700,112 |
| Unix | 16 | 3.2 | 3,949,373 | 4,923,380 | 181,120 |
| Mixed | 4 | 0.8 | 1,184,521 | 1,420,492 | 417,792 |
| Windows | 3 | 0.6 | 465,600 | 628,129 | 46,092 |
| BSD Based | 1 | 0.2 | 122,400 | 131,072 | 1,280 |

Fuente: <http://top500.org/statistics/list/>

GNU/Linux



GNU/Linux



¿Por qué Linux es mejor?

<http://www.whylinuxisbetter.net/>



Olvida los virus.



¿Tu sistema es inestable?



Linux protege tu ordenador.



No pagues 100 dólares por tu sistema operativo.



No más mi**daware.



¡Libertad!



Una vez instalado tu sistema, ¿Por qué aún necesitas instalar más cosas?



Olvida los controladores (drivers).



Actualiza todo tu software con un sólo clic.



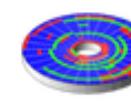
¿Por qué hacer copias ilegales de software, si puedes tenerlo gratis?



¿Necesitas software nuevo? No te moleste buscando en la web, Linux lo hace por ti.



Entra en la nueva generación de escritorios.



¿Tu vida digital parece fragmentada?



Escoge la apariencia de tu escritorio.



¿Por qué tu Windows es más lento cada día?



Haz algo por el entorno.



Sin puertas traseras en tu software.



Disfruta de soporte gratis e ilimitado.



¿Demasiadas ventanas? Usa las distintas áreas de trabajo.



Sin desastres en tu menú principal.



No esperes años a que se arreglen los fallos, comunícalos y sigue su solución.



¿Estás cansado de reiniciar tu ordenador continuamente?



Deja que tu viejo ordenador tenga una segunda vida.



Juega gratis a cientos de juegos.

Los pocos casos donde deberías quedarte con Windows (por ahora)



Existe un programa propietario sin el que no puedes vivir



Eres un jugador empedernido



Trabajas para la industria de impresión de libros



Tu hardware no está aún soportado

¿Quiere instalar GNU/Linux?

Si está interesado en instalar GNU/Linux, le recomiendo instalar la distribución Linux Mint (con el escritorio KDE): <http://www.linuxmint.com/>

Se recomienda hacerlo inicialmente dentro de una máquina virtual (si tiene al menos 4 Gb de RAM):

https://www.youtube.com/results?search_query=instalar+linux+mint+18+virtualbox

De lo contrario instale el sistema operativo desde el Live CD (no hay necesidad de borrar Windows para tal fin):

https://www.youtube.com/results?search_query=instalar+linux+mint+18+live+DVD

Bytes

| Representations | | | | Customary prefix | | |
|-----------------|-----------|-------|-----------------------------------|-------------------------------|--------|------|
| Base 2 | Base 1024 | Value | Base 10 | Name | Symbol | |
| 2^{10} | 1024^1 | | 1 024 | $\approx 1.02 \times 10^3$ | kilo | k, K |
| 2^{20} | 1024^2 | | 1 048 576 | $\approx 1.05 \times 10^6$ | mega | M |
| 2^{30} | 1024^3 | | 1 073 741 824 | $\approx 1.07 \times 10^9$ | giga | G |
| 2^{40} | 1024^4 | | 1 099 511 627 776 | $\approx 1.10 \times 10^{12}$ | tera | T |
| 2^{50} | 1024^5 | | 1 125 899 906 842 624 | $\approx 1.13 \times 10^{15}$ | peta | P |
| 2^{60} | 1024^6 | | 1 152 921 504 606 846 976 | $\approx 1.15 \times 10^{18}$ | exa | E |
| 2^{70} | 1024^7 | | 1 180 591 620 717 411 303 424 | $\approx 1.18 \times 10^{21}$ | zetta | Z |
| 2^{80} | 1024^8 | | 1 208 925 819 614 629 174 706 176 | $\approx 1.21 \times 10^{24}$ | yotta | Y |

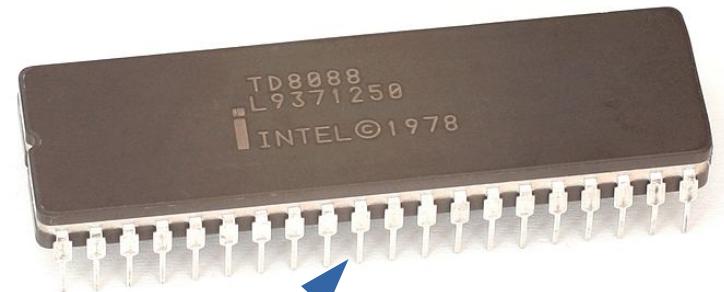
Lenguajes de programación

Es un conjunto de instrucciones diseñadas para expresar procesos que pueden ser llevados a cabo por las computadoras:

- **De máquina:** lo entiende directamente el procesador; dichas instrucciones ejercen un control directo sobre el hardware y están condicionados por la estructura física de la computadora que lo soporta. Unos y ceros.
- **De bajo nivel (lenguaje ensamblador):** forma pnemotécnica de representar el lenguaje de máquina; constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador.
- **De nivel intermedio**
- **De alto nivel:** expresan los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas.

Lenguaje de máquina

| | | |
|------------|---|--|
| -u 100 1a | OCFD :0100 BA0B01 OCFD :0103 B409 OCFD :0105 CD21 OCFD :0107 B400 OCFD :0109 CD21 | MOV DX,010B MOV AH,09 INT 21 MOV AH,00 INT 21 |
| -d 10b 13f | OCFD :0100 20 65 73 74 65 20 65 73-20 75 48 6F 6C 61 2C OCFD :0110 72 61 6D 61 20 68 65 63-68 6F 20 65 6E 20 61 73 OCFD :0120 73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20 OCFD :0130 57 69 6B 69 70 65 64 69-61 24 | Este es un programa hecho en assembler para la Wikipedia\$ |



Hola,
este es un progra
ma hecho en as
sembler para la
Wikipedia\$

Lenguaje de máquina del **Intel 8088**. El código de máquina en hexadecimal se resalta en rojo, el equivalente en **lenguaje ensamblador** en magenta, y las direcciones de memoria donde se encuentra el código, en azul. Abajo se ve un texto en hexadecimal y ASCII.

Lenguaje de bajo nivel

Ventajas:

- Máximo aprovechamiento de los recursos de la máquina.
- Máxima velocidad posible.

Desventajas:

- Mínima portabilidad (depende de la máquina)
- Requiere de la máxima atención y de una organización estructurada en base a los planos del hardware y del objetivo del software.
- Largo tiempo de desarrollo
- Difícil encontrar errores

Lenguaje ensamblador

The screenshot shows a window titled "file:///home/diego/programas/hello.asm - Kate". The menu bar includes File, Edit, Document, View, Bookmarks, Tools, Sessions, Settings, Window, and Help. The toolbar contains icons for new file, open file, save file, cut, copy, paste, find, and search. A vertical sidebar on the left is titled "Filesystem Browser" and shows a tree view of the file structure. The main text area contains the following assembly code:

```
1 section .data
2     hello:    db 'Hola a todos!!!',10 ; 'Hola a todos!!! mas el caracter \n
3     helloLen: equ $-hello           ; Longitud de la cadena 'Hola a todos!!!'
4
5 section .text
6     global _start
7
8 _start:
9     mov eax,4          ; Llamado al sistema para escribir
10    mov ebx,1           ; 1 quiere decir que se escribirá en la pantalla
11    mov ecx,hello      ; Se especifica la cadena a escribir y su longitud
12    mov edx,helloLen   ;
13    int 80h             ; Se llama al kernel
14
15    mov eax,1          ; Se llama al sistema para salir
16    mov ebx,0           ; Se sale con código 0 (no error)
17    int 80h
```

The status bar at the bottom shows "Line: 12 Col: 26" and the file path "file:///home/diego/programas/hello.asm". Below the editor, a terminal window displays the following command-line session:

```
diego@earendil:~/programas$ nasm -f elf hello.asm
diego@earendil:~/programas$ ls
hello.asm  hello.asm~  hello.o
diego@earendil:~/programas$ ld -s -o HelloWorld_en_ASM hello.o
diego@earendil:~/programas$ ls
hello.asm  hello.asm~  hello.o  HelloWorld_en_ASM*
diego@earendil:~/programas$ ./HelloWorld_en_ASM
Hola a todos!!
```

At the very bottom, there are "Find in Files" and "Terminal" buttons.

Lenguaje de nivel intermedio

The screenshot shows a window titled "file:///home/diego/programas/hello.c - Kate". The menu bar includes File, Edit, Document, View, Bookmarks, Tools, Sessions, Settings, Window, and Help. The toolbar contains icons for file operations like Open, Save, Print, and Find. On the left, there's a "Documents" sidebar and a "Filesystem Browser" with a tree view. The main code editor area displays the following C program:

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     printf("Hola a todos!!\n");
6     return 0;
7 }
```

The status bar at the bottom shows "Line: 6 Col: 13" and the file path "file:///home/diego/programas/hello.c". Below the editor, a terminal window shows the following session:

```
diego@earendil:~/programas$ gcc -o HelloWorld_en_C
diego@earendil:~/programas$ ls
hello.c  HelloWorld_en_C*
diego@earendil:~/programas$ ./HelloWorld_en_C
Hola a todos!!
diego@earendil:~/programas$ 
```

At the bottom, there are "Find in Files" and "Terminal" buttons.

Session Edit View Bookmarks Settings Help

```
diego@earrendil:~/programas_C$ gdb HelloWorld_en_C
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
(gdb) disassemble main
Dump of assembler code for function main:
0x080483a4 <main+0>:    lea    0x4(%esp),%ecx
0x080483a8 <main+4>:    and    $0xffffffff,%esp
0x080483ab <main+7>:    pushl  -0x4(%ecx)
0x080483ae <main+10>:   push   %ebp
0x080483af <main+11>:   mov    %esp,%ebp
0x080483b1 <main+13>:   push   %ecx
0x080483b2 <main+14>:   sub    $0x4,%esp
0x080483b5 <main+17>:   movl   $0x8048490,(%esp)
0x080483bc <main+24>:   call   0x80482d4 <puts@plt>
0x080483c1 <main+29>:   mov    $0x0,%eax
0x080483c6 <main+34>:   add    $0x4,%esp
0x080483c9 <main+37>:   pop    %ecx
0x080483ca <main+38>:   pop    %ebp
0x080483cb <main+39>:   lea    -0x4(%ecx),%esp
0x080483ce <main+42>:   ret
End of assembler dump.
(gdb) █
```



Shell



Lenguajes de alto nivel

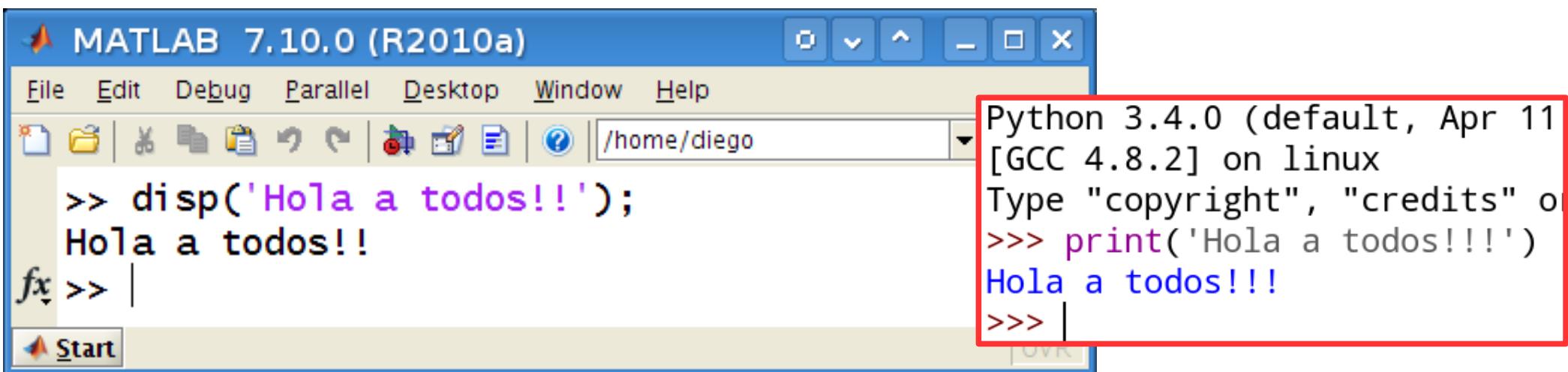
Ventajas

- Genera un código más sencillo y comprensible.
- Escribir un código válido para diversas máquinas y, posiblemente, sistemas operativos.

Inconvenientes

- Reducción de velocidad al ceder el trabajo de bajo nivel a la máquina.
- Algunos requieren que la máquina cliente posea una determinada plataforma.

Lenguaje de alto nivel



A screenshot of the Microsoft Excel environment and the Microsoft Visual Basic Editor.

The Excel window (Libro1) shows a blank sheet with columns A, B, C, D and rows 1, 2, 3, 4, 5, 6, 7. Cell A1 is selected.

The Microsoft Visual Basic - Libro1 [ejecutando] window shows the following code in the Immediate window:

```
MsgBox("Hola a todos!")
```

A message box titled "Microsoft Excel" is displayed, containing the text "Hola a todos!!!" and an "Aceptar" button.

The Project Explorer window shows the following structure:

- atpybaen.xls (ATPYBAE)
- VBAProject (FUNCRES.X)
- VBAProject (Libro1)
 - Microsoft Excel Objetos
 - Hoja1 (Hoja1)
 - Hoja2 (Hoja2)
 - Hoja3 (Hoja3)
 - ThisWorkbook

Visual Basic for Applications: Alt+F11 en MS EXCEL

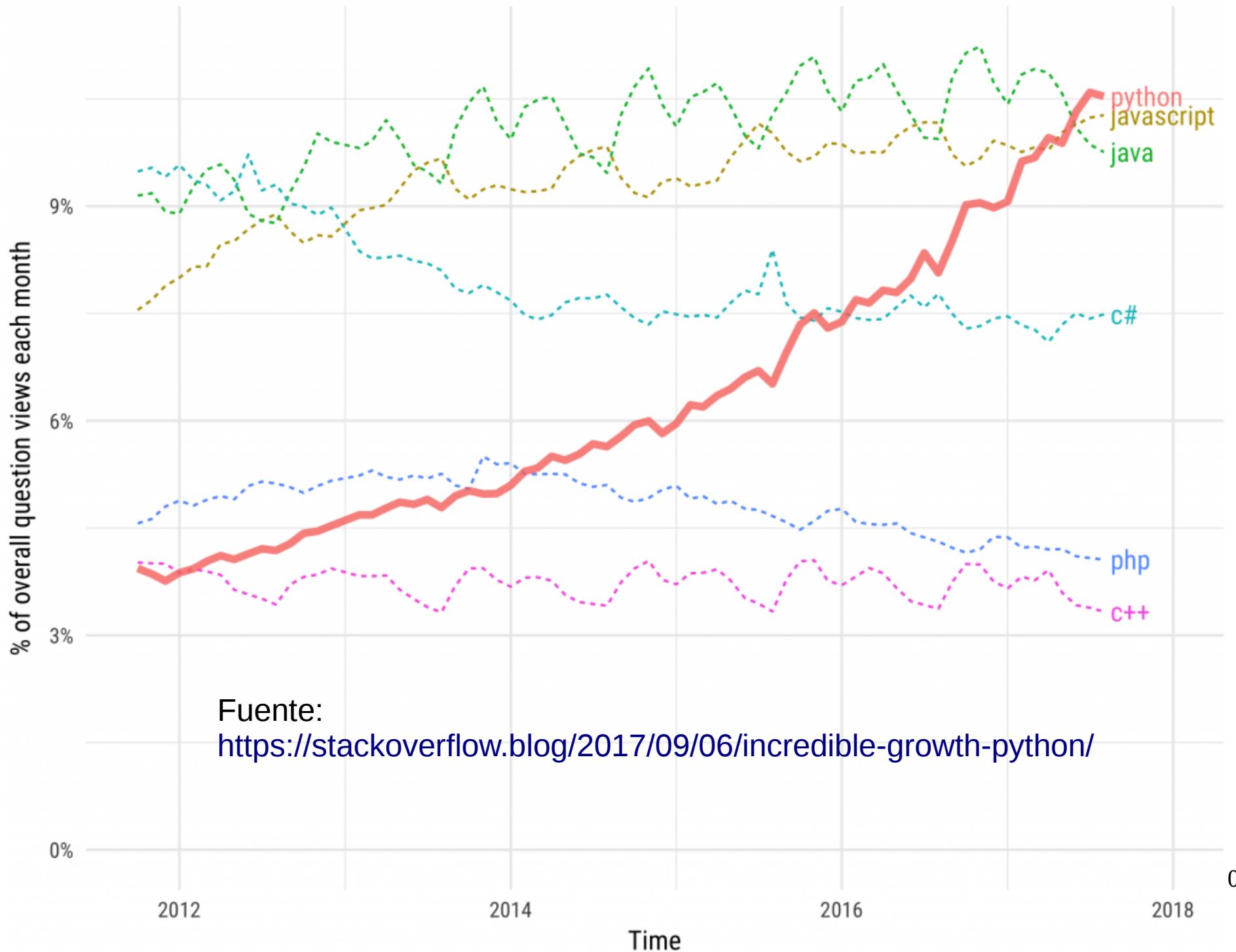
Python vs. otros lenguajes (Agosto 2018)

FUENTE: <https://www.tiobe.com/tiobe-index/>

| Aug 2018 | Aug 2017 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 16.881% | +3.92% |
| 2 | 2 | | C | 14.966% | +8.49% |
| 3 | 3 | | C++ | 7.471% | +1.92% |
| 4 | 5 | ▲ | Python | 6.992% | +3.30% |
| 5 | 6 | ▲ | Visual Basic .NET | 4.762% | +2.19% |
| 6 | 4 | ▼ | C# | 3.541% | -0.65% |
| 7 | 7 | | PHP | 2.925% | +0.63% |
| 8 | 8 | | JavaScript | 2.411% | +0.31% |
| 9 | - | ▲ | SQL | 2.316% | +2.32% |
| 10 | 14 | ▲ | Assembly language | 1.409% | -0.40% |
| 11 | 11 | | Swift | 1.384% | -0.44% |
| 12 | 12 | | Delphi/Object Pascal | 1.372% | -0.45% |
| 13 | 17 | ▲ | MATLAB | 1.366% | -0.25% |
| 14 | 18 | ▲ | Objective-C | 1.358% | -0.15% |
| 15 | 10 | ▼ | Ruby | 1.182% | -0.78% |
| 16 | 9 | ▼ | Perl | 1.175% | -0.82% |
| 17 | 16 | ▼ | Go | 0.996% | -0.65% |
| 18 | 15 | ▼ | R | 0.965% | -0.80% |
| 19 | 13 | ▼ | Visual Basic | 0.922% | -0.89% |

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Fuente:

<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

Python



Creado por **Guido van Rossum** (**NL, 1956**). Se empezó a implementar en Dic. 1989 como un pasatiempo para realizar durante las vacaciones de navidad.

Python 0.9.0 (Feb 1991)

Python 1.0 (Ene. 1994)

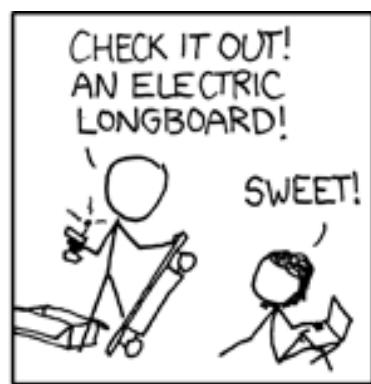
Python 2.0 (Oct. 2000)

Python 3.0 (Dic. 2008)

El nombre está inspirado en el programa de la BBC “Monty Python’s Flying Circus”. Nada que ver con serpientes.

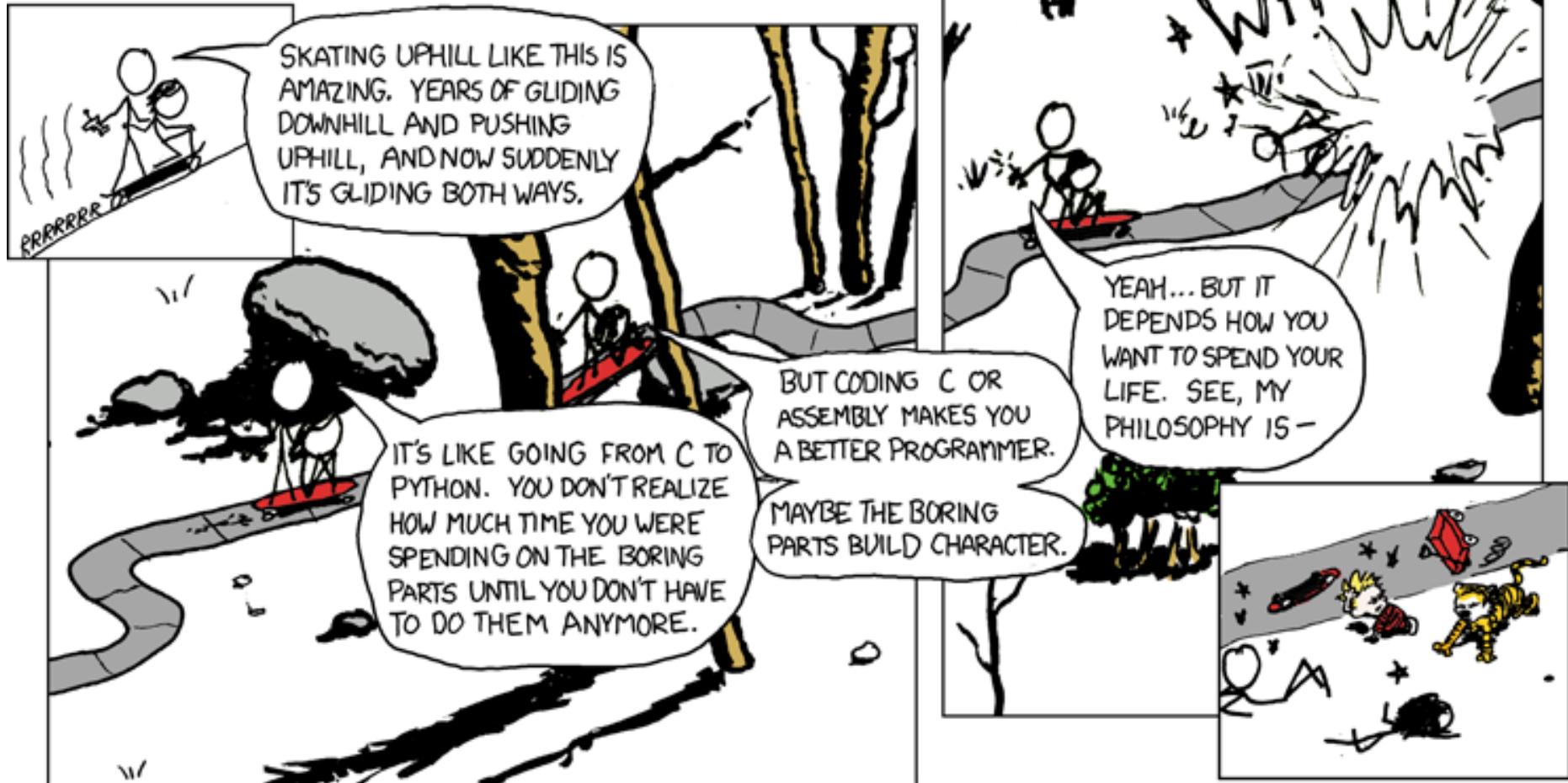
Características de Python

- Es uno de los lenguajes de programación más fáciles de aprender (es un lenguaje de muy alto nivel).
- Es un lenguaje interpretado.
- Lenguaje que permite la creación de prototipos rápidamente. Permite el desarrollo de programas complejos en poco tiempo y en pocas líneas comparado con C/C++ o Java.
- El código fuente es muy fácil de leer y entender (es prácticamente pseudocódigo ejecutable).
- Se pueden crear funciones hechas en C o C++ y llamarlas desde Python.
- Permite programación funcional y orientada a objetos.



“Python vs C”

Tomado de: <http://xkcd.com/409/>



Python 2 vs. Python 3

- Python 3 (2008) no es compatible con Python 2 (2000). Python 2 no se actualiza desde el 2010. Solo se le corrigen errores de programación y errores críticos de seguridad. Se soportará hasta el 2020.
- Python 3 corrige muchas inconsistencias que tiene Python 2, haciéndolo mucho más fácil de aprender para el principiante.
- Sin embargo Python 2 sigue siendo muy utilizado ya que existen numerosas librerías escritas de excelente calidad escritas en ese lenguaje y que aun no han sido traducidas.
- En el primer semestre de 2013 por primera vez se empezó a descargar más Python 3 que Python 2.

¿Qué es un compilador?

Un **compilador** es un programa que permite traducir el **código fuente** de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior (típicamente **lenguaje de máquina**).

De esta manera un programador puede diseñar un programa en un lenguaje mucho más cercano a como piensa un ser humano, para luego compilarlo a un programa más manejable por una computadora.

¿Qué es un intérprete?

El **intérprete** o **interpretador** es un programa informático capaz de analizar y ejecutar otros programas. Los intérpretes se diferencian de los compiladores o de los ensambladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema, los intérpretes sólo realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.

Los intérpretes son muy útiles ya que son buenos para experimentar con diferentes instrucciones sin tener que correr el programa, lo cual disminuye el tiempo de desarrollo considerablemente, haciendo la programación más interactiva.

Compilador vs. Intérprete

- Un **compilador** lee completamente un programa en lenguaje de nivel intermedio/alto y lo traduce en su integridad a un programa de código de máquina equivalente. El programa de código de máquina resultante se puede ejecutar cuantas veces se desee, sin necesidad de volver a traducir el programa original. Ejemplo: lenguaje C, C++, Pascal, etc.
- Un **intérprete** lee un programa escrito en un lenguaje de alto nivel instrucción a instrucción y, para cada una de ellas, efectúa una traducción a las instrucciones de código de máquina equivalentes y las ejecuta inmediatamente. No hay un proceso de traducción separado por completo del de ejecución. Cada vez que ejecutamos el programa con un intérprete, se repite el proceso de traducción y ejecución, ya que ambos son simultáneos. Esto hace que el lenguaje interpretado sea entre 10 y 30 veces más lento que el lenguaje compilador. Ejemplo: MATLAB, Python, VisualBasic.

Implementaciones de Python (intérpretes)

- **C_Python** (esta es la implementación oficial y la que utilizaremos)
- Brython
- CLPython
- IronPython
- Jython
- PyMite
- PyPy
- RapydScript
- SNAPpy
- Tinypy

Entre otras: ver <https://wiki.python.org/moin/PythonImplementations>

Implementaciones de Python (compiladores)

- Psyco
- 2c-python
- Compyler
- Cython
- GCC Python Front-End
- Numba
- Pyc
- Shed Skin
- UnPython

Entre otras: <https://wiki.python.org/moin/PythonImplementations>

C~~P~~ython

C~~P~~ython es la implementación oficial y más ampliamente utilizada del lenguaje de programación Python. Está escrita en C.

Está desarrollada por varios desarrolladores de software libre, la comunidad usuaria de Python y la "Python Software Foundation".

Se descarga de: <https://www.python.org/downloads/>

Instalación en Linux:

```
# sudo apt-get install python3 idle3
```

Versiones actuales de CPython

- Python 2.7: tendrá soporte hasta el 2020 (bugfixes).
- Python 3 (3.7 es la versión más reciente y la que veremos en el curso)
- Ambas versiones se descargan de <https://www.python.org/downloads/>

32 o 64 bits?

- Si usted tiene Windows de 32 bits, instale la versión de 32 bits.
- Si usted tiene más de 4 Gb de memoria RAM y Windows de 32, cambie su sistema operativo por uno de 64 bits.
- En un sistema operativo de 64 bits se puede instalar Python de 32 o de 64 bits.
- Algunos programadores dicen que para evitar ciertos problemas de compatibilidad entre paquetes se recomienda instalar Python de 32 bits.

El intérprete de Python

- En Linux:
 - /usr/bin/python (Python 2.7)
 - /usr/bin/python3 (Python 3.x)
- Windows:
 - C:\Python37\

```
daa@heimdall ~ $ python
Python 2.7.6 (default, Mar 22 2014, 22:59:38)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
daa@heimdall ~ $ python3
Python 3.4.0 (default, Apr 11 2014, 13:05:18)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
daa@heimdall ~ $ █
```

Modo interactivo

- Se caracteriza por el cursor símbolo
`>>>`
- Sirve como calculadora
- Sirve para comprobar el contenido de las variables en memoria
- Sirve para ejecutar pequeños pedazos de código y mirar como se comportan

Shells para trabajo interactivo

- Python3 (the standard python interpreter)
- idle3
- jupyter (antes llamado ipython)
<https://jupyter.org/> <https://ipython.org/>
 - console
 - qtconsole
 - notebook

El archivo de código fuente en un lenguaje interpretado se le conoce como **script**. Este tiene extensión **.py**

Aquí se observa la forma de ejecutar el script.

Se sale con:

- `exit()` o `quit()`
- Linux: `Ctrl + D`
- Windows: `Ctrl + Z`

The screenshot shows a terminal window with the following content:

```
01_hola_mundo.py
1 # Este programa pregunta su nombre y lo saluda
2 nombre = input('¿Cuál es tu nombre? ')
3 print('Gusto en conocerlo, ' + nombre + '.')
Line 4, Column 1
daalvarez@eredron ~ $ python3 01_hola_mundo.py
¿Cuál es tu nombre? PEPITO
Gusto en conocerlo, PEPITO.
daalvarez@eredron ~ $ python3
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('01_hola_mundo.py').read())
¿Cuál es tu nombre? PEPITO
Gusto en conocerlo, PEPITO.
>>> nombre
'PEPITO'
>>>
daalvarez@eredron ~ $ jupyter console
Jupyter console 5.1.0

Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- 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]: %run 01_hola_mundo.py
¿Cuál es tu nombre? PEPITO
Gusto en conocerlo, PEPITO.

In [2]: nombre
Out[2]: 'PEPITO'

In [3]:
Do you really want to exit ([y]/n)? y
Shutting down kernel
daalvarez@eredron ~ $
```

The terminal shows the execution of a Python script named `01_hola_mundo.py`. It asks for the user's name and prints a greeting. It also demonstrates running the script from the command line, using the `jupyter console`, and displaying help information. A large curly brace on the right side groups the terminal output into two modes: "Modo programa autónomo" (Mode autonomous program) for the first three lines and "Modo interactivo" (Interactive mode) for the rest of the session.

Documentación

- La documentación oficial (en PDF):
<https://docs.python.org/3.6/download.html>
- En linux:
`# sudo apt-get install python3-doc python3.6-doc python3-examples`
- La ayuda en línea con help()

```
>>> help(pow)
Help on built-in function pow in module builtins:

pow(...)
    pow(x, y[, z]) -> number

    With two arguments, equivalent to x**y.  With three arguments,
    equivalent to (x**y) % z, but may be more efficient (e.g. for ints).

>>> |
```

Interactive help

```
>>> help
Type help() for interactive help, or help(object) for help about object.
>>> help()
```

Welcome to Python 3.4! This is the interactive help utility.

If this is your first time using Python, you should definitely check out the tutorial on the Internet at <http://docs.python.org/3.4/tutorial/>.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", type "modules spam".

```
help> pow
Help on built-in function pow in module builtins:
```

```
pow(...)
    pow(x, y[, z]) -> number
```

With two arguments, equivalent to $x**y$. With three arguments, equivalent to $(x**y) \% z$, but may be more efficient (e.g. for ints).

```
help> |
```

Interactive help in jupyter

Jupyter QtConsole 4.3.0
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- 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]: `pow?`

Signature: `pow(x, y, z=None, /)`

Docstring:

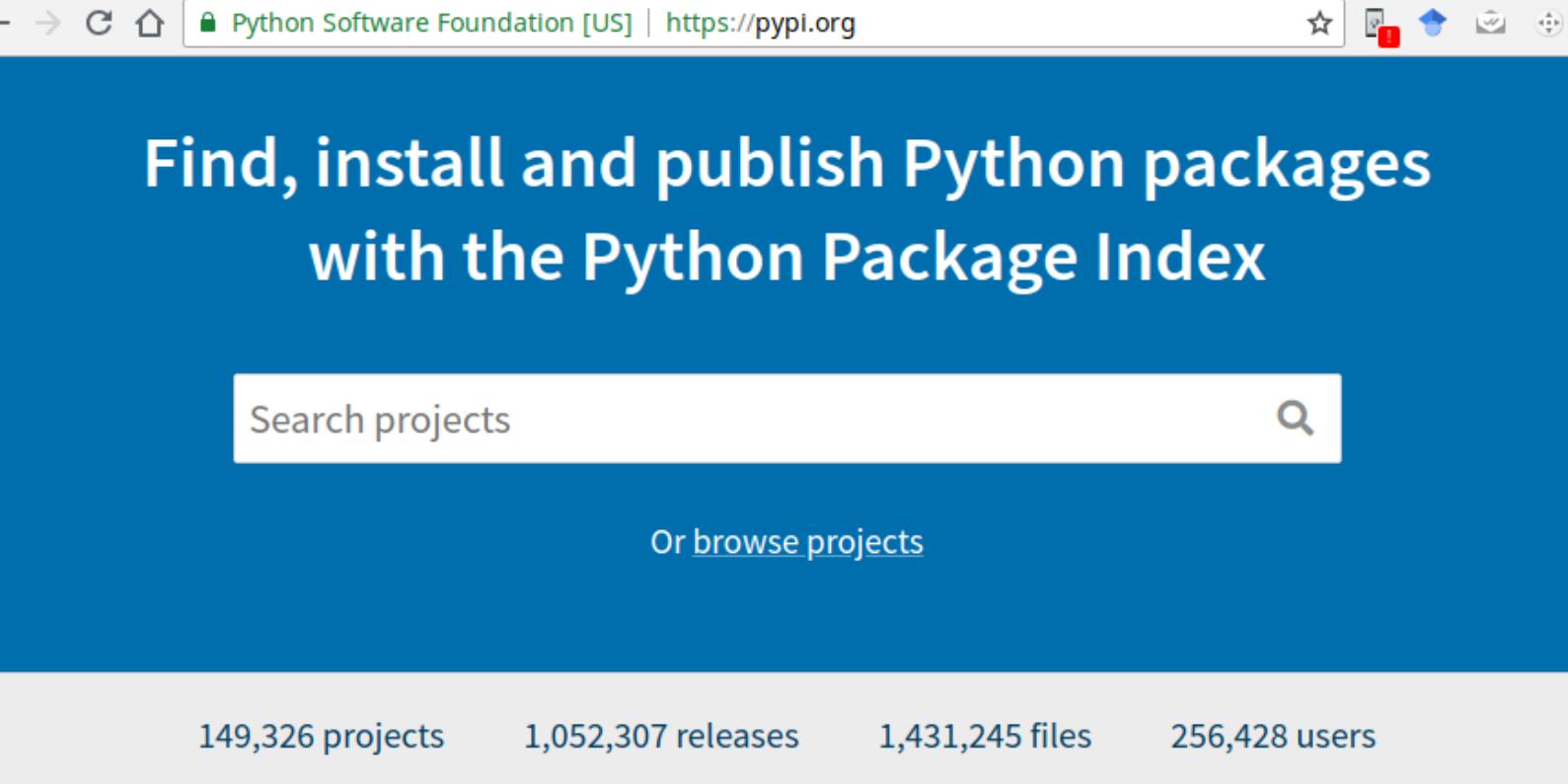
Equivalent to `x**y` (with two arguments) or `x**y % z` (with three arguments)

Some types, such as ints, are able to use a more efficient algorithm when invoked using the three argument form.

Type: `builtin_function_or_method`

In [2]:

PyPI – The Python Package Index



The screenshot shows the main landing page of PyPI. At the top, there's a browser header with the URL "Python Software Foundation [US] | https://pypi.org". Below the header is a large blue banner with white text that reads "Find, install and publish Python packages with the Python Package Index". Underneath the banner is a search bar containing the placeholder "Search projects" with a magnifying glass icon. Below the search bar is a link "Or browse projects". At the bottom of the main content area, there are four statistics: "149,326 projects", "1,052,307 releases", "1,431,245 files", and "256,428 users".



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages.](#)

Este es un repositorio de librerías de Python hechas por personas como usted o yo. Actualmente alberga 149.326 proyectos (Agosto 17, 2018).

Distribuciones de Python (basadas en CPython)

- ActivePython from ActiveState
- **Anaconda** from Continuum Analytics
<https://www.anaconda.com/distribution/>
- Enthought's Canopy
- PocketPython
- PyIMSL Studio
- PyPy: a Python implementation in Python.
- StacklessPython
- entre otras...

Esta es la distribución de Python que utilizaremos y que debemos instalar



Creando código en Python

- Intérprete
- Se utiliza cualquier editor de texto (extensión **.py**). Procure que la codificación de sus archivos de texto sea UTF-8.
- El editor de moda en el momento es **Visual Studio Code**
<https://code.visualstudio.com/>
- Otro editor de texto bastante popular entre los phytonistas es **Sublime Text**
<http://www.sublimetext.com/>

Codificación de los archivos

- Python 2.7: ASCII
- Python 3: Unicode UTF-8

Si usted trabaja en
Windows ¡actívela!

Si usted quiere utilizar una codificación diferente a la UTF-8, en la primera o segunda línea de su script, escriba:

```
# -*- coding: iso-8859-1 -*-
```

Coloque aquí el nombre
de su codificación.

Ver codificaciones disponibles en: <https://docs.python.org/3.4/library/codecs.html>

Entorno de desarrollo integrado (IDE Integrated Development Environment)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en:

- un editor de código
- un compilador/interpretador
- un depurador
- un constructor de interfaz gráfica (GUI)

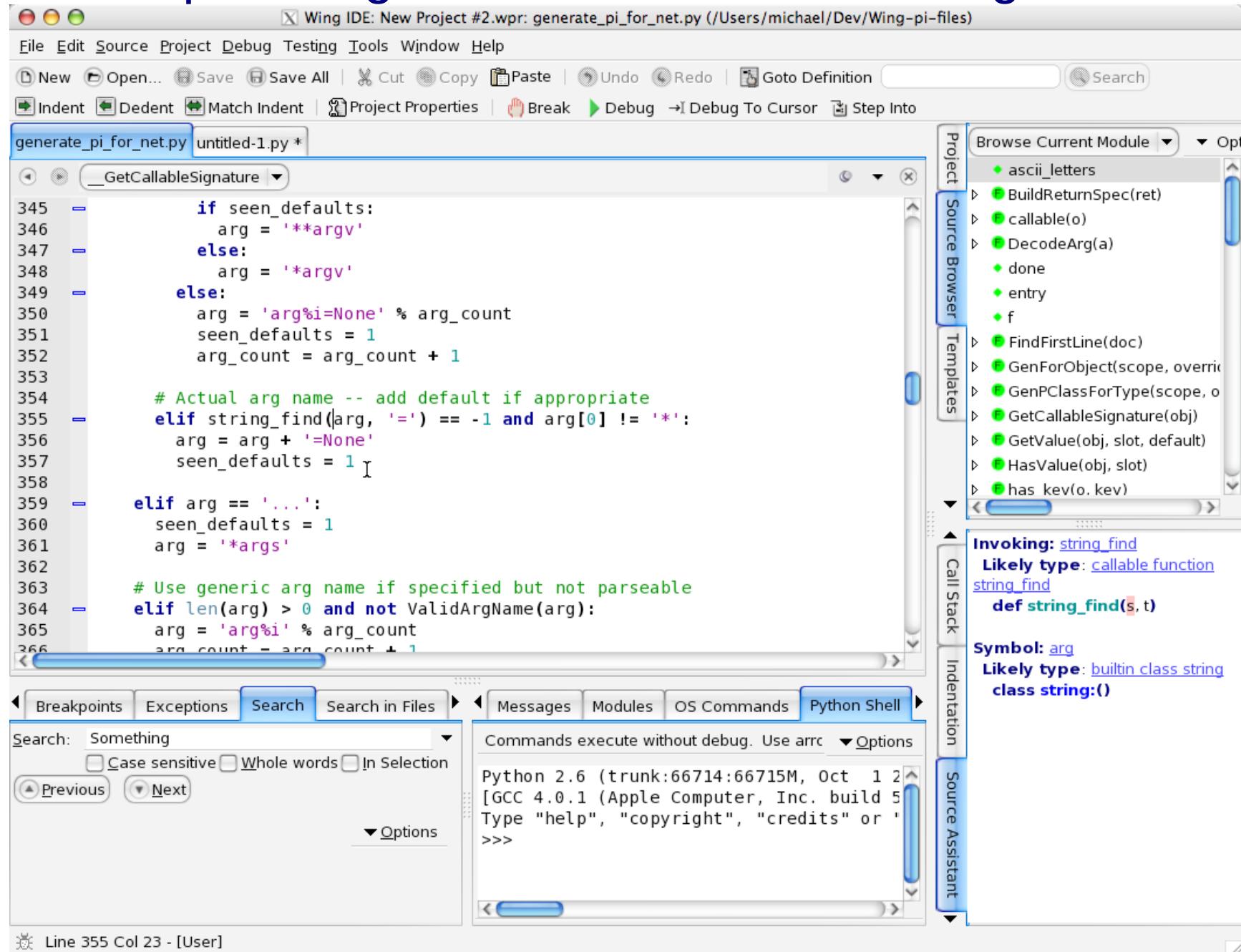
IDEs para Python

- Se recomienda:
 - Wing IDE 101 <http://wingware.com/downloads/wingide-101>
 - PyCharm Educational Edition <https://www.jetbrains.com/pycharm-educational/>
 - Spyder (viene incluído en Anaconda) <https://github.com/spyder-ide/spyder>
- Otros IDEs son:
 - PyDev + Eclipse
 - Komodo IDE
 - Ninja

Procure que la codificación de sus
archivos de texto sea UTF-8.

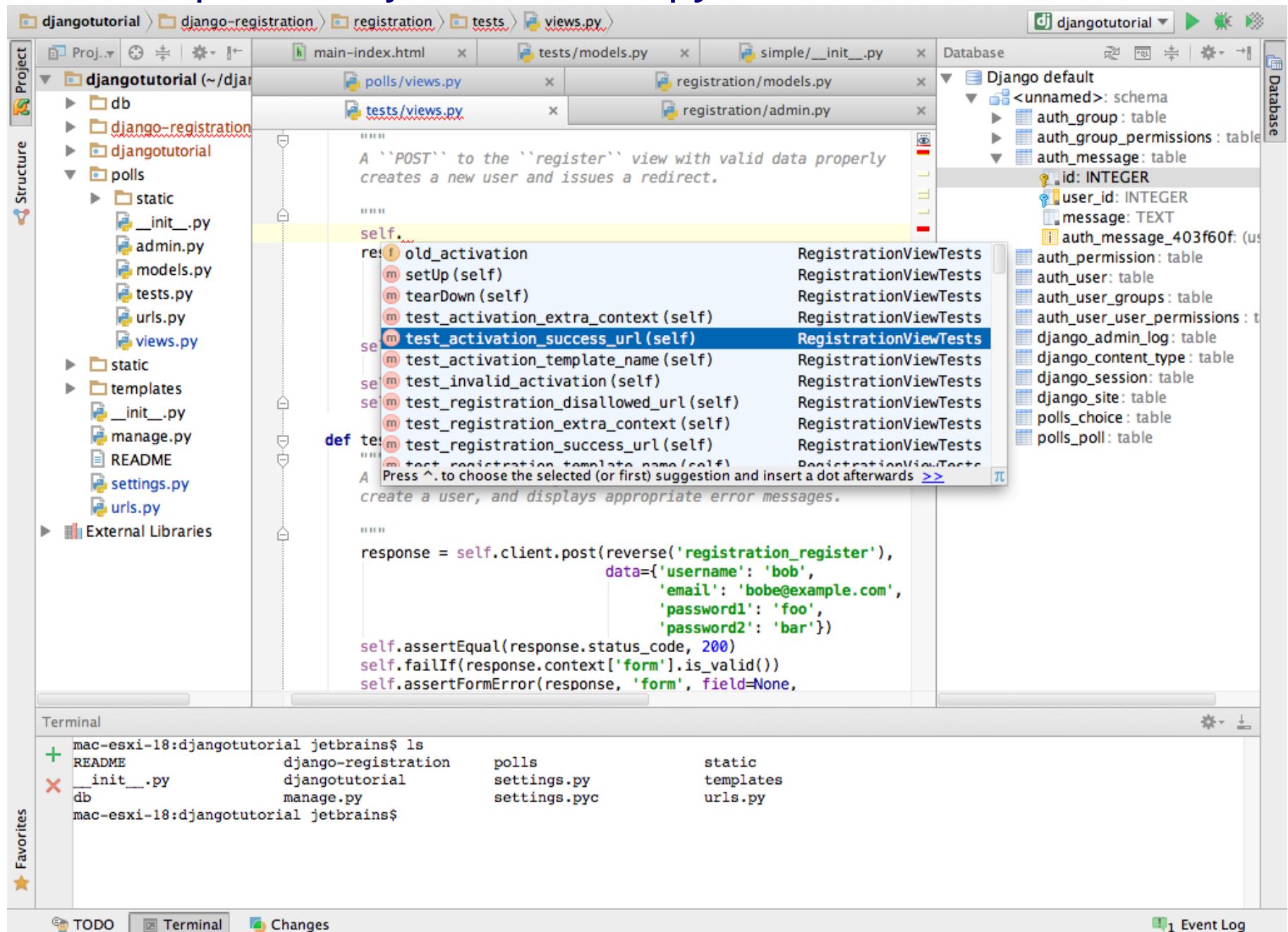
Wing IDE

<https://wingware.com/downloads/wing-101>



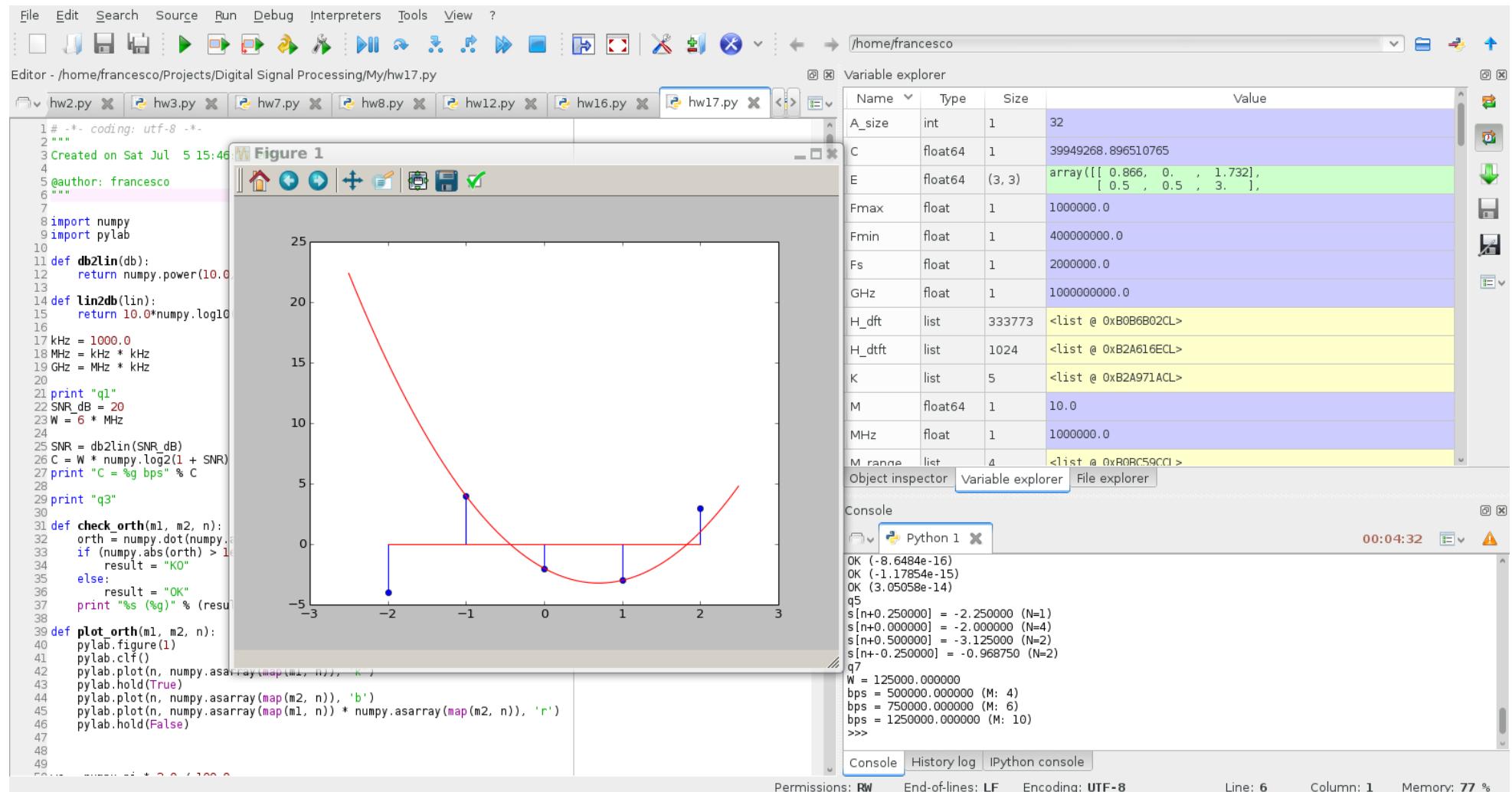
PyCharm

<https://www.jetbrains.com/pycharm-edu/download/>



Spyder

Viene incluído con el Anaconda <https://www.spyder-ide.org/>

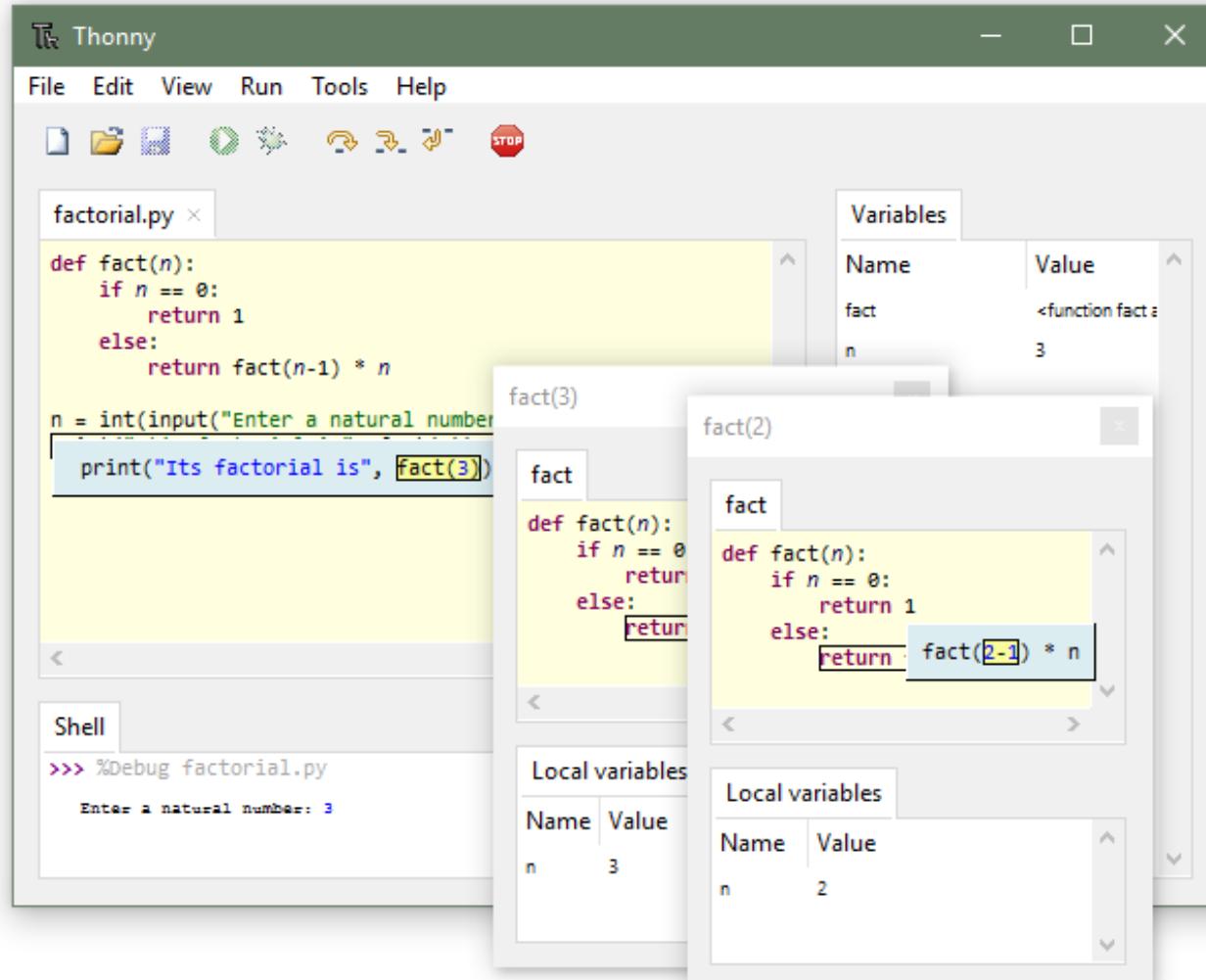
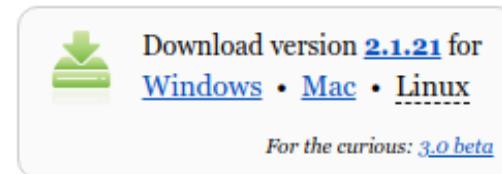


Thonny

<https://thonny.org/>

Thonny

Python IDE for beginners

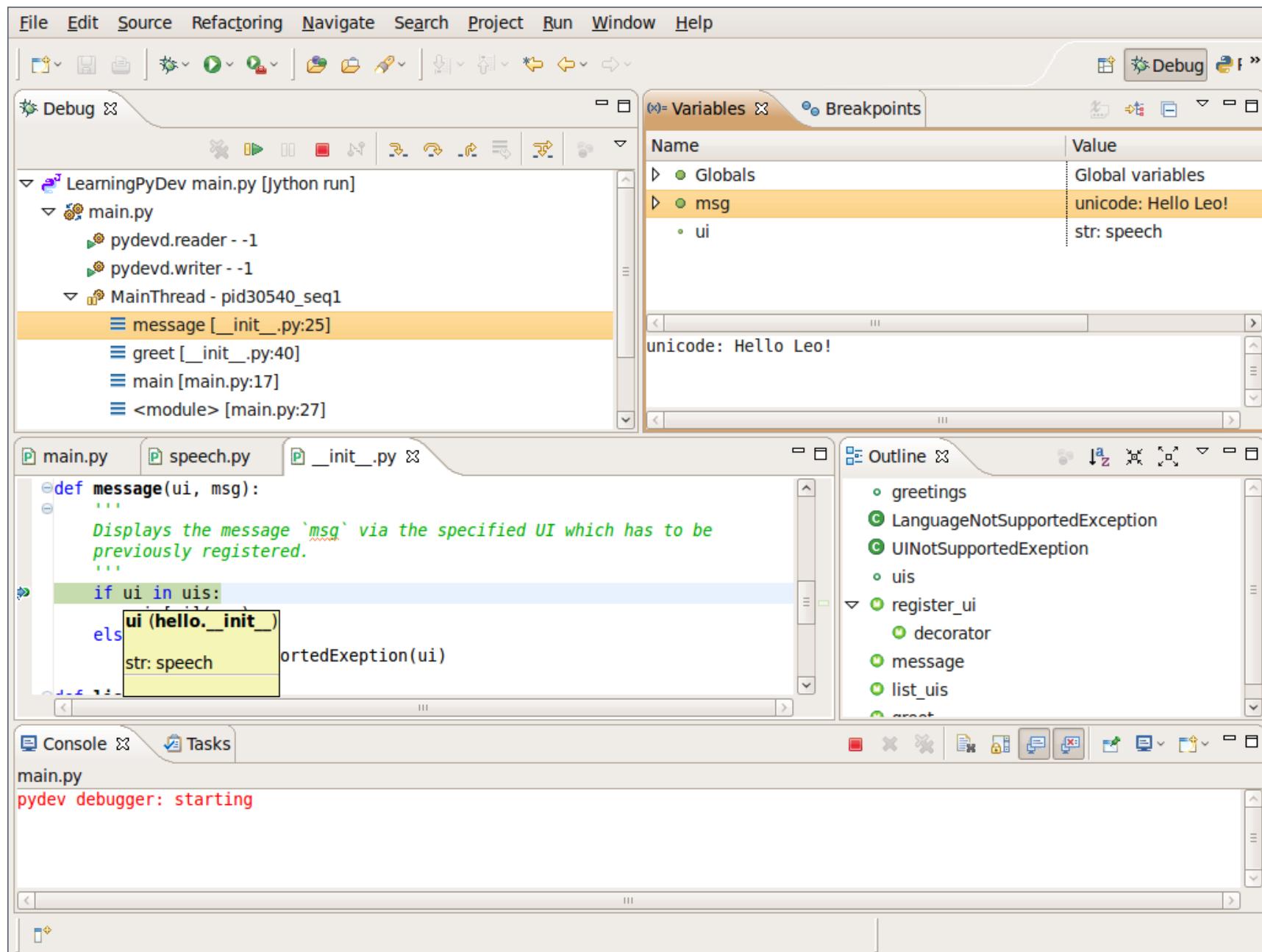


Komodo

The screenshot shows the Komodo Edit 6.0 interface. The main window has a title bar "edile-0.2.py (~/Downloads) - Komodo Edit 6.0". The menu bar includes File, Edit, Code, Navigation, View, Project, Tools, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. On the left is a "Places" sidebar with a tree view of local files and a "Projects" section. The central area has four tabs: "Start Page", "prefs.xml", "drm_buffer.h", and "edile-0.2.py" (the active tab). The "edile-0.2.py" tab displays Python code for a GTK application. Below the tabs is a status bar showing "Ln: 1 Col: 1" and "14/12/2010 16:41:46". The right side features a "Source" sidebar with a tree view of methods and their implementations, such as "print_about()", "on_about_menu_item_act...", and "show_about()". At the bottom, there's a footer with links to "Archive", "By email", and "List of historical anniversaries".

```
1212     # filename or None.
1213     def get_save_filename(self):
1214         filename = None
1215         chooser = gtk.FileChooserDialog("Save File...", self.window,
1216                                         gtk.FILE_CHOOSER_ACTION_SAVE,
1217                                         (gtk.STOCK_CANCEL, gtk.RESPONSE_CANCEL,
1218                                          gtk.STOCK_SAVE, gtk.RESPONSE_OK))
1219
1220         response = chooser.run()
1221         if response == gtk.RESPONSE_OK: filename = chooser.get_filename()
1222         #if response == gtk.RESPONSE_CANCEL: print "save cancelled"
1223         chooser.destroy()
1224
1225         return filename
1226
1227
1228     """
1229     # do_* = controller methods called from on_* actions
1230     """
1231
1232     # find
1233
1234     def get_find_iter(self,buffer=None, backwards=False,limit_iter=None):
1235         find_iter = None
1236         selection = buffer.get_selection_bounds()
1237         if selection == ():
1238             find_iter = buffer.get_iter_at_mark(buffer.get_insert())
1239             if find_iter == None:
1240                 find_iter = buffer.get_bounds()[int(backwards)]
1241             else:
1242                 find_iter = selection[int(not backwards)]
```

PyDev



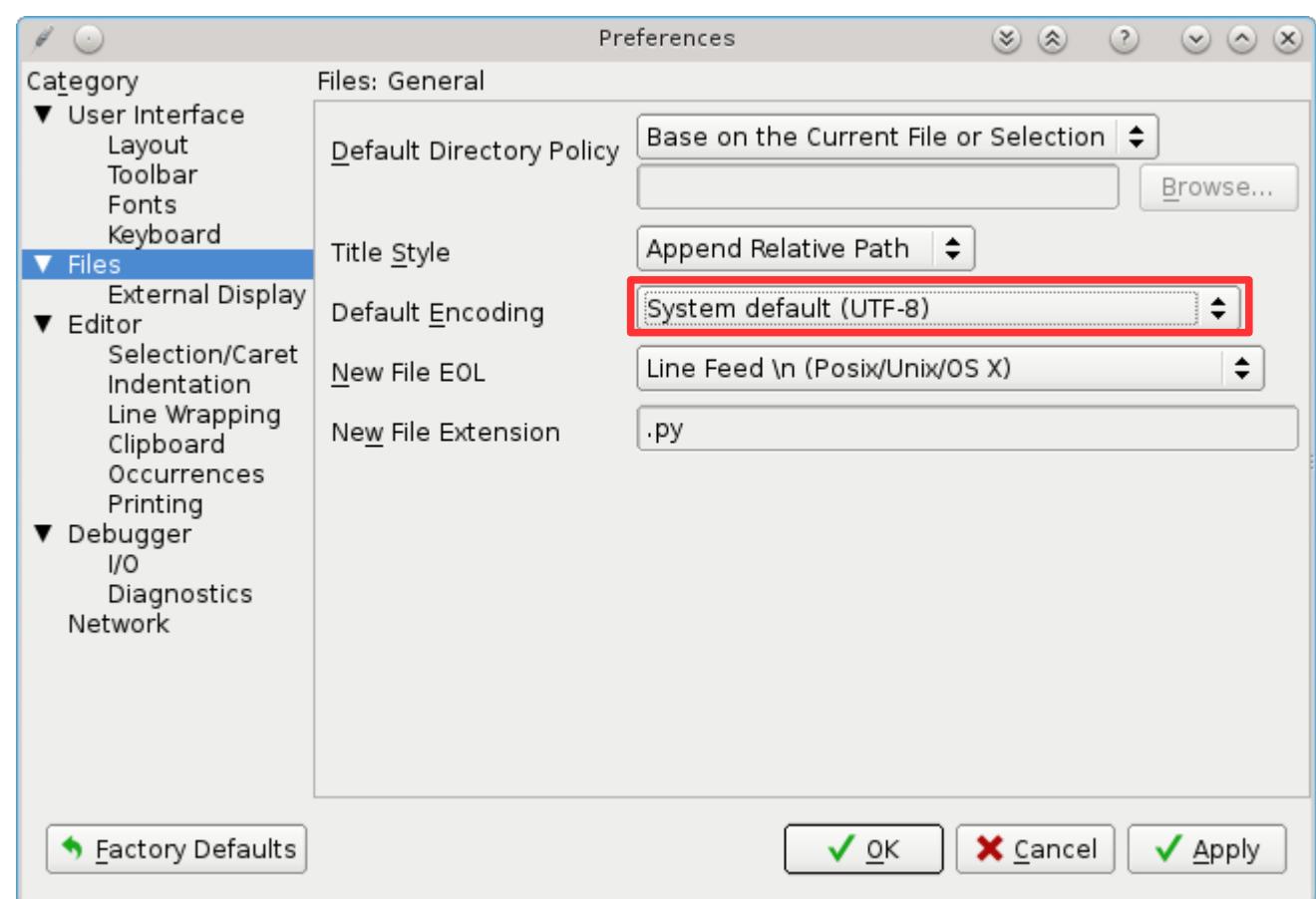
Activando la codificación UTF-8

WingIDE 101:

Edit -> Preferences

Files -> Default Encoding

Unicode (UTF - 8)



Referencias

- Wikipedia
- <http://www.inventwithpython.com/>
- <http://www.diveintopython3.net/>
- Documentación de Python:
 - <https://docs.python.org/3/tutorial/index.html>
 - <https://docs.python.org/3/>
- Marzal Varó, Andrés; Gracia Luengo, Isabel; García Sevilla, Pedro (2014). Introducción a la programación con Python 3. Disponible en: <http://dx.doi.org/10.6035/Sapientia93>

Consejos

- Aprender a programar es una tarea que requiere perseverancia y paciencia.
- Estudiar con juicio el código fuente de otras personas es un excelente método de aprendizaje ya que así se descubren trucos y formas de programar.
- Aprenda a frecuentar los foros de programadores en internet (por ejemplo: <https://stackoverflow.com/>). Allí encontrará cientos de personas que lo pueden ayudar a resolver sus dudas.

Referencias

- Wikipedia
- <http://www.inventwithpython.com/>
- <http://www.diveintopython3.net/>
- Documentación de Python:
 - <https://docs.python.org/3/tutorial/index.html>
 - <https://docs.python.org/3/>
- Marzal Varó, Andrés; Gracia Luengo, Isabel; García Sevilla, Pedro (2014). Introducción a la programación con Python 3. Disponible en: <http://dx.doi.org/10.6035/Sapientia93>