



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencias de la Computación

## Clase 02: Instalación Python

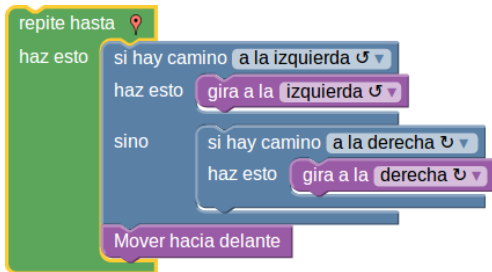
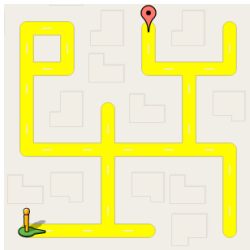
Rodrigo Toro Icarte (rntoro@uc.cl)

IIC1103 Introducción a la Programación - Sección 5

09 de Marzo, 2015

# Control sorpresa!

- 1) ¿Cuál es la clave para pasar el ramo? Hint: P\_\_\_\_\_
- 2) ¿Defina en sus palabras qué es un algoritmo?
- 3) Diga qué ocurrirá



# Python

- Lenguaje de programación.
- Usaremos versión 3.x

# Python Ubuntu

Seguramente ya tienen instalado python.

Abrir Terminal y ejecutar:

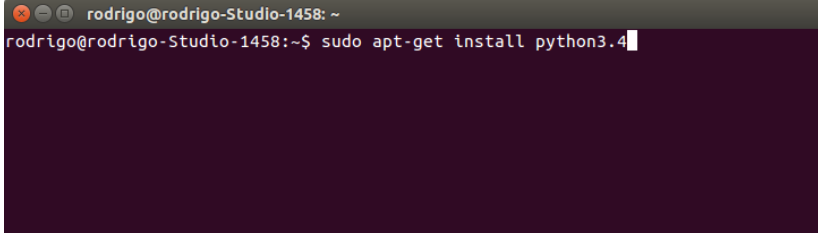
**python3 --version**

```
rodriego@rodriego-Studio-1458: ~  
rodriego@rodriego-Studio-1458:~$ python3 --version  
Python 3.4.0  
rodriego@rodriego-Studio-1458:~$
```

# Python Ubuntu

Si no está instalado ejecutar:

```
sudo apt-get install python3.4
```



A terminal window with a dark background. The title bar shows "rodrigo@rodrigo-Studio-1458: ~". The terminal prompt is "rodrigo@rodrigo-Studio-1458:~\$". The command "sudo apt-get install python3.4" is entered and followed by a white cursor. There is a small orange vertical bar at the bottom right corner of the terminal window.

```
rodrigo@rodrigo-Studio-1458: ~  
rodrigo@rodrigo-Studio-1458:~$ sudo apt-get install python3.4
```



# Python Ubuntu

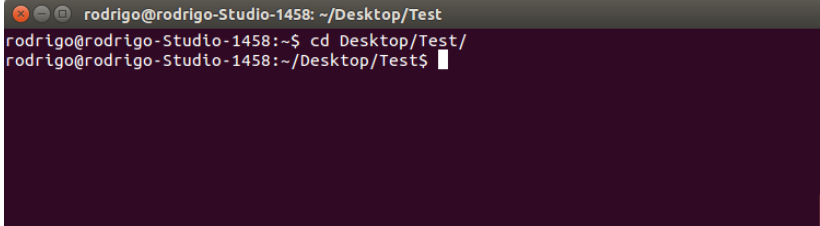
Abrir documento y escribir:

```
print("Hello world!")
```

A screenshot of a gedit editor window on Ubuntu. The window title is "example.py (~/Desktop/Introducción a la Programación/Test) - gedit". The menu bar includes "Open", "Save", "Print", "Undo", and "Cut". The main editing area shows a single line of Python code: `print("Hello world!")`. The status bar at the bottom indicates "Python", "Tab Width: 2", "Ln 1, Col 22", and "INS".

# Python Ubuntu

En Terminal ir a la carpeta mediante **cd**:  
**cd Desktop/Test/**

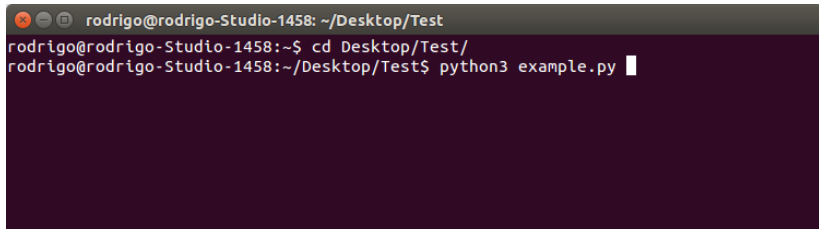


```
rodrigo@rodrigo-Studio-1458: ~/Desktop/Test
rodrigo@rodrigo-Studio-1458:~$ cd Desktop/Test/
rodrigo@rodrigo-Studio-1458:~/Desktop/Test$
```



# Python Ubuntu

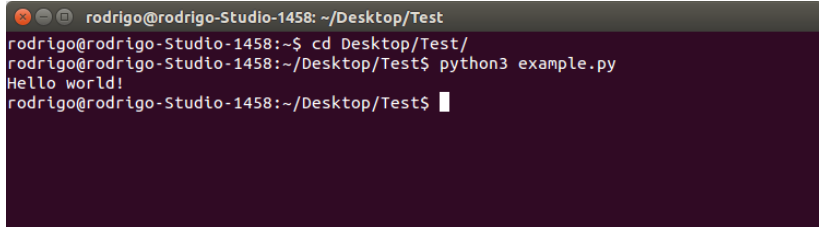
Ejecutar el programa con:  
**python3 example.py**

A terminal window with a dark background and light text. The title bar shows 'rodrigo@rodrigo-Studio-1458: ~/Desktop/Test'. The terminal content shows the user navigating to the Desktop/Test directory and running the command 'python3 example.py'.

```
rodrigo@rodrigo-Studio-1458: ~/Desktop/Test
rodrigo@rodrigo-Studio-1458:~$ cd Desktop/Test/
rodrigo@rodrigo-Studio-1458:~/Desktop/Test$ python3 example.py
```

# Python Ubuntu

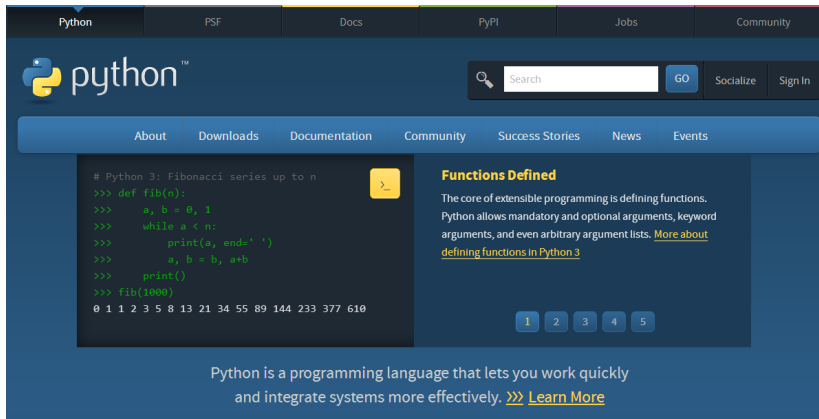
Ejecutar el programa con:  
**python3 example.py**

A terminal window with a dark background and light text. The window title bar shows 'rodrigo@rodrigo-Studio-1458: ~/Desktop/Test'. The terminal content shows the user navigating to the Desktop/Test directory and running 'python3 example.py', which outputs 'Hello world!' and returns the prompt.

```
rodrigo@rodrigo-Studio-1458: ~/Desktop/Test
rodrigo@rodrigo-Studio-1458:~$ cd Desktop/Test/
rodrigo@rodrigo-Studio-1458:~/Desktop/Test$ python3 example.py
Hello world!
rodrigo@rodrigo-Studio-1458:~/Desktop/Test$ █
```

# Instalar Python (Windows y Mac)

www.python.org



The screenshot shows the Python.org website interface. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo and a search bar with a 'GO' button. A secondary navigation bar contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a code editor on the left with a Python 3 script for calculating the Fibonacci series up to n. The script defines a function fib(n) and prints the first 1000 numbers of the series. The output shows the sequence: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610. To the right of the code editor is a section titled 'Functions Defined' which explains that the core of extensible programming is defining functions, and provides a link to 'More about defining functions in Python 3'. Below this text are five numbered buttons (1-5). At the bottom of the page, a message states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```

### Functions Defined

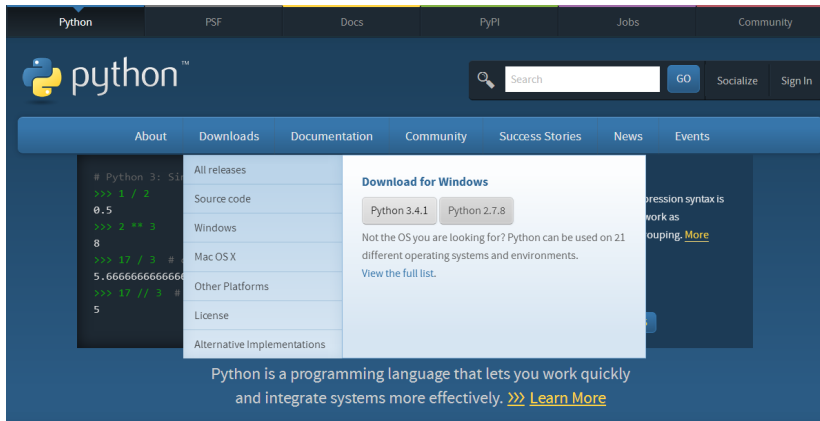
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

# Instalar Python (Windows y Mac)

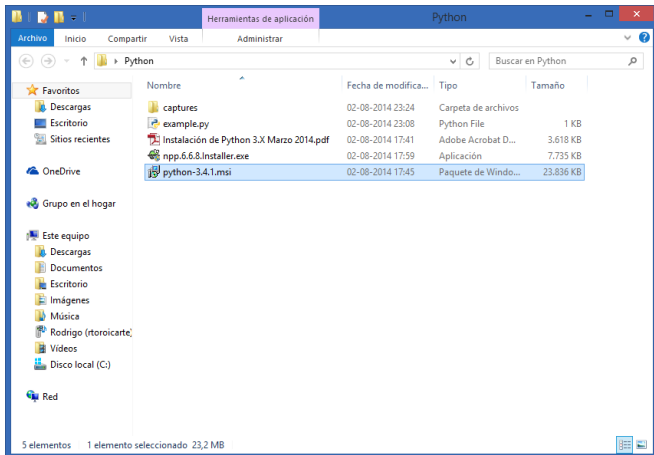
[www.python.org](http://www.python.org) → Download → Python 3.X



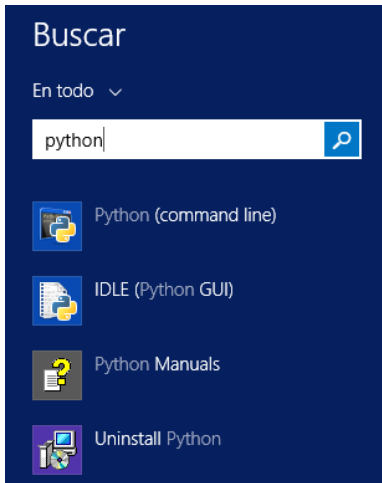
The screenshot shows the Python.org website interface. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo and a search bar. A secondary navigation bar includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The 'Downloads' menu is open, showing options for All releases, Source code, Windows, Mac OSX, Other Platforms, License, and Alternative Implementations. The 'Download for Windows' section is highlighted, featuring buttons for Python 3.4.1 and Python 2.7.8. Below these buttons, text states: 'Not the OS you are looking for? Python can be used on 21 different operating systems and environments. [View the full list.](#)' At the bottom of the page, a message reads: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'

# Instalar Python (Windows y Mac)

Ejecutar descarga... siguiente, siguiente, siguiente, y fin!



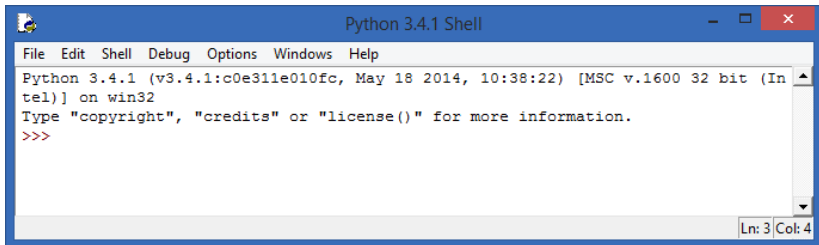
# Instalar Python (Windows y Mac)



¿Qué se instaló?

- 1 Python (command line)
- 2 IDLE (Python GUI)

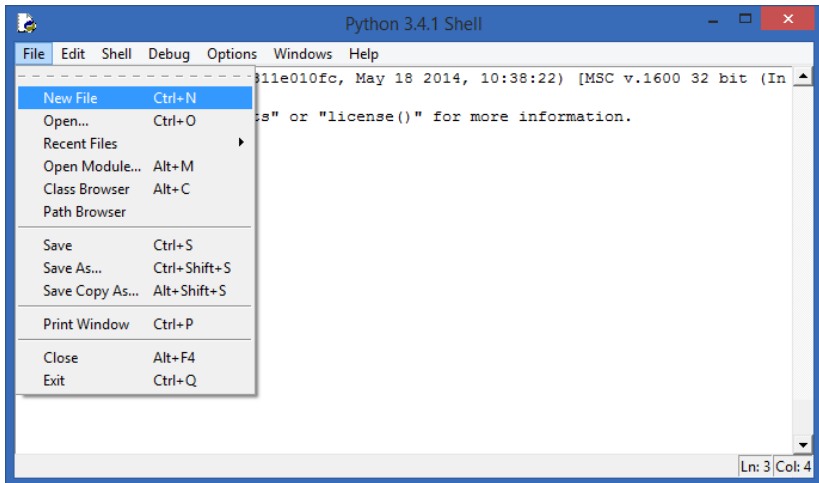
# IDLE



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

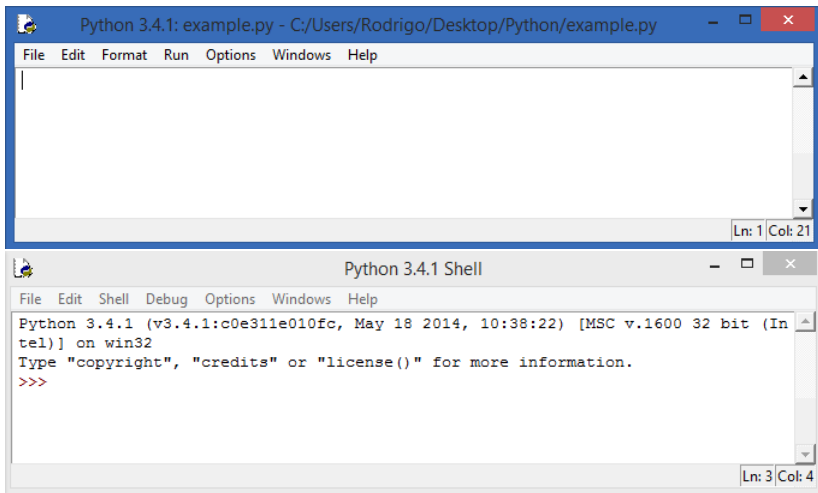
Ln: 3 Col: 4

# IDLE (File → New File)

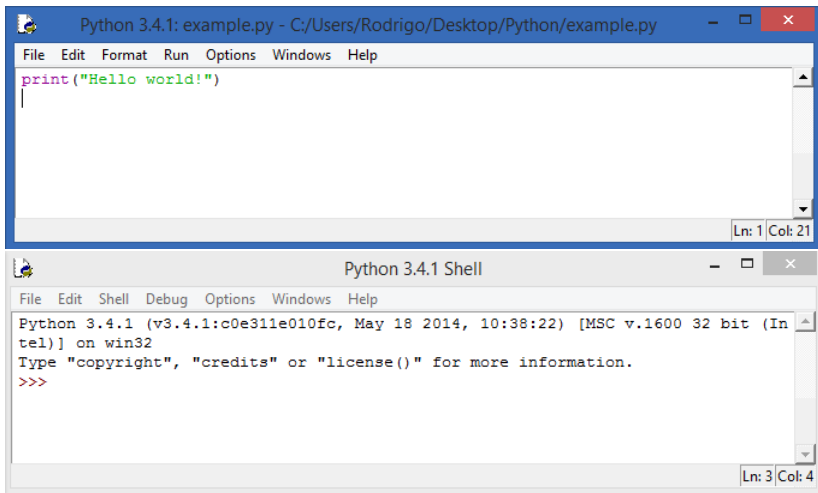




# IDLE (escribir programa)



# IDLE (escribir programa)

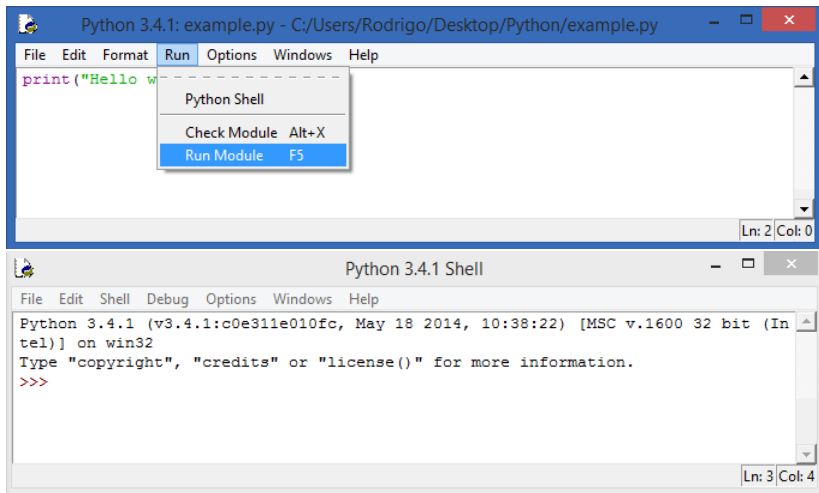


The screenshot displays the Python IDLE interface. The top window, titled "Python 3.4.1: example.py - C:/Users/Rodrigo/Desktop/Python/example.py", contains a single line of Python code: `print("Hello world!")`. The bottom window, titled "Python 3.4.1 Shell", shows the output of running the script, including the Python version and build information, and the prompt `>>>`.

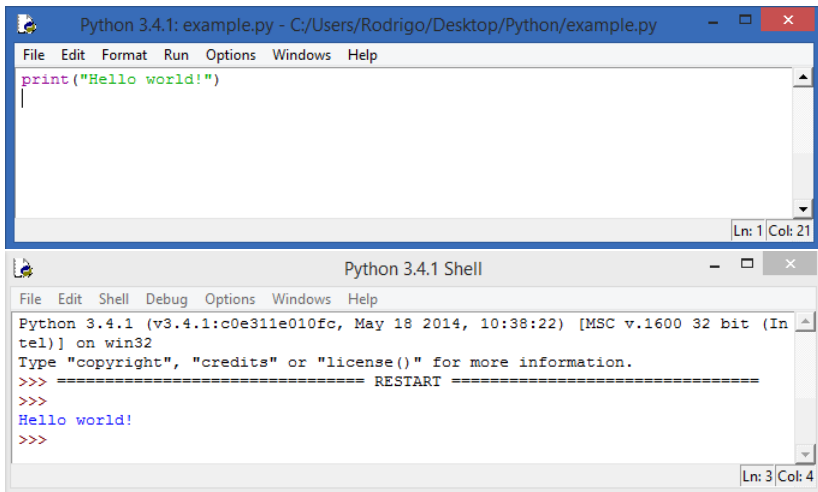
```
Python 3.4.1: example.py - C:/Users/Rodrigo/Desktop/Python/example.py
File Edit Format Run Options Windows Help
print("Hello world!")
Ln: 1 Col: 21

Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
Ln: 3 Col: 4
```

# IDLE (Run → Run Module)









# IDLE (resultados en consola)



The image shows two windows from the Python IDLE environment. The top window is the Python 3.4.1 editor, titled "Python 3.4.1: example.py - C:/Users/Rodrigo/Desktop/Python/example.py". It contains a single line of Python code: `print("Hello world!")`. The bottom window is the "Python 3.4.1 Shell", which shows the output of running the script. The shell text is as follows:

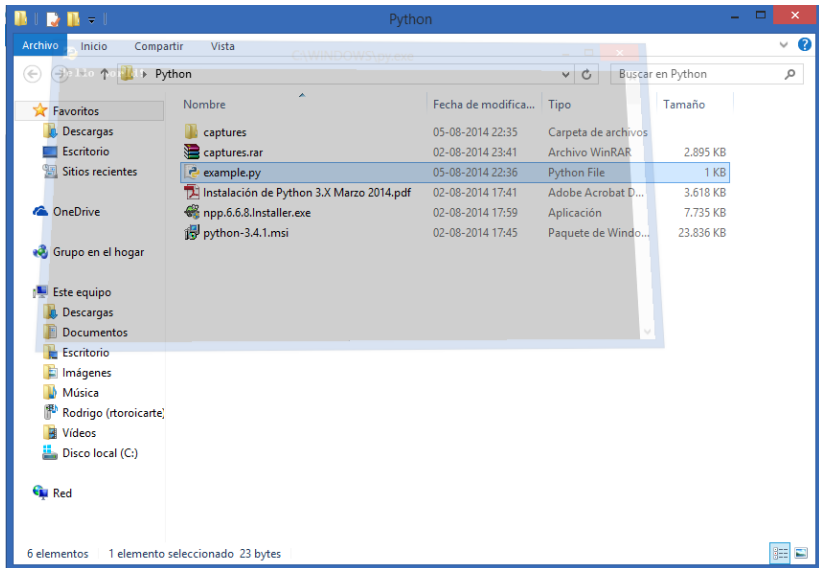
```
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello world!
>>>
```

# ¿Qué guardamos?

 captures	02-08-2014 23:40	Carpeta de archivos	
 captures.rar	02-08-2014 23:41	Archivo WinRAR	2.895 KB
 example.py	02-08-2014 23:39	Python File	1 KB
 Instalación de Python 3.X Marzo 2014.pdf	02-08-2014 17:41	Adobe Acrobat D...	3.618 KB
 npp.6.6.8.Installer.exe	02-08-2014 17:59	Aplicación	7.735 KB
 python-3.4.1.msi	02-08-2014 17:45	Paquete de Windo...	23.836 KB

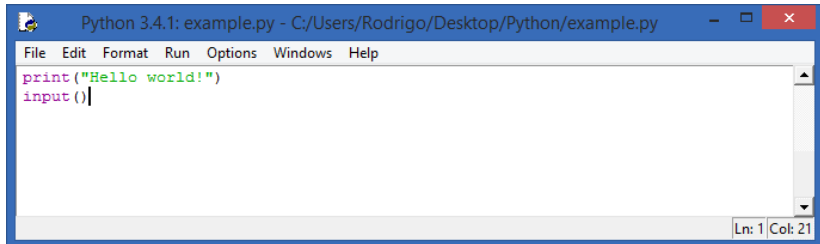
Obs: example.py no es más que un archivo de texto.

# Doble-click en example.py



# Doble-click en example.py

Para evitar que el programa se cierre usamos comando **input()**.



The screenshot shows a window titled "Python 3.4.1: example.py - C:/Users/Rodrigo/Desktop/Python/example.py". The window contains a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following code:

```
print("Hello world!")  
input()
```

The status bar at the bottom right of the window indicates "Ln: 1 Col: 21".

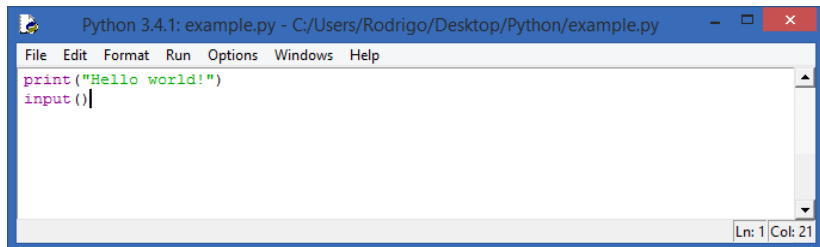
# Doble-click en example.py





# Editores de texto

El editor de texto por defecto de python es algo incómodo...



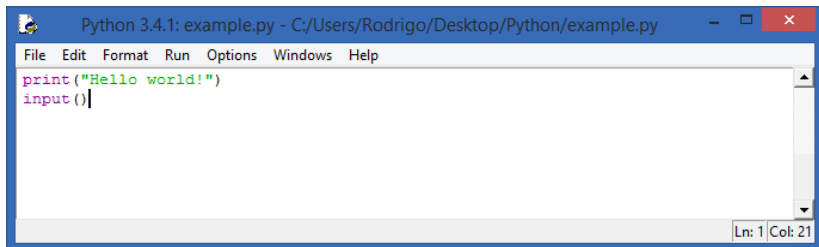
The screenshot shows a window titled "Python 3.4.1: example.py - C:/Users/Rodrigo/Desktop/Python/example.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
print("Hello world!")  
input()
```

The status bar at the bottom right indicates "Ln: 1 | Col: 21".

# Editores de texto

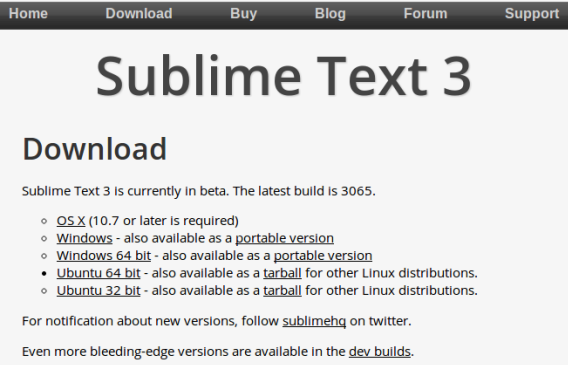
El editor de texto por defecto de python es algo incómodo...



Les recomiendo usar **Sublime Text 3**.

# Sublime Text

<http://www.sublimetext.com/3>



The screenshot shows the top navigation bar of the Sublime Text website with links for Home, Download, Buy, Blog, Forum, and Support. Below the navigation bar is the main heading "Sublime Text 3" and a sub-heading "Download". The text states that Sublime Text 3 is currently in beta and provides the latest build number, 3065. A list of operating systems and their respective download options is provided, including OS X, Windows (with portable versions), and Ubuntu (with tarball options). It also mentions following @sublimehq on Twitter for notifications and that dev builds are available for those seeking bleeding-edge versions.

Home Download Buy Blog Forum Support

## Sublime Text 3

### Download

Sublime Text 3 is currently in beta. The latest build is 3065.

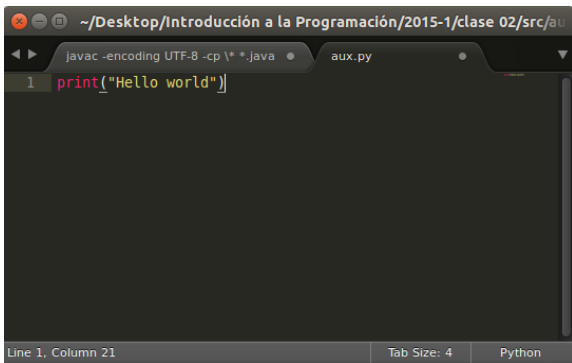
- [OS X](#) (10.7 or later is required)
- [Windows](#) - also available as a [portable version](#)
- [Windows 64 bit](#) - also available as a [portable version](#)
- [Ubuntu 64 bit](#) - also available as a [tarball](#) for other Linux distributions.
- [Ubuntu 32 bit](#) - also available as a [tarball](#) for other Linux distributions.

For notification about new versions, follow [@sublimehq](#) on twitter.

Even more bleeding-edge versions are available in the [dev builds](#).

# Sublime Text

Ahora podemos editar nuestro código desde Sublime Text.



The screenshot shows the Sublime Text 3 interface. The window title is `~/Desktop/Introducción a la Programación/2015-1/clase 02/src/au`. There are two tabs: `javac -encoding UTF-8 -cp \* *.java` and `aux.py`. The `aux.py` tab is active, showing a single line of Python code: `1 print("Hello world")`. The status bar at the bottom indicates `Line 1, Column 21`, `Tab Size: 4`, and `Python`.

# Blockly Code

Blockly Code permite programar con bloques (**link**).

## Blockly : Code



# Blockly Code

Tiene manejo de texto (mostrar mensaje, pedir datos)

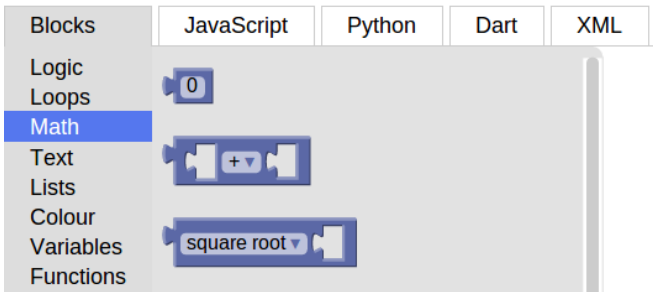
## Blockly : Code

The image shows the Blockly code editor interface. On the left is a vertical sidebar with a list of block categories: Blocks, Logic, Loops, Math, Text (highlighted in blue), Lists, Colour, Variables, and Functions. On the right, there are four tabs: JavaScript, Python, Dart, and XML. Below the tabs, a workspace contains several code blocks. At the top is a green block with a speech bubble icon and the text "“ ”". Below it is a green block with the text "print". At the bottom is a green block with the text "prompt for text with message" and a dropdown menu showing "text", followed by a green block with a speech bubble icon and the text "“ ”".

# Blockly Code

Tiene operaciones matemáticas (+-\*/)

## Blockly : Code



# Blockly Code

Ejemplo:

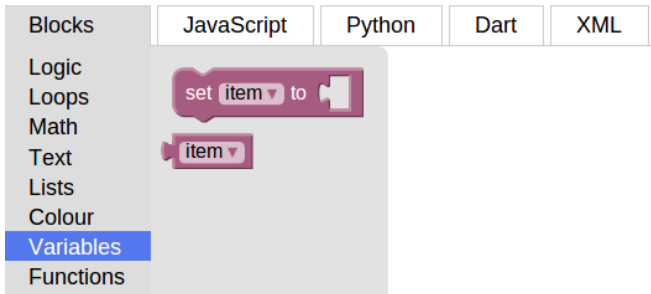
The image shows a screenshot of the Blockly code editor interface. On the left side, there is a vertical sidebar with a list of categories: Blocks, Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. The 'Text' category is currently selected. At the top of the main workspace, there are four tabs: JavaScript, Python, Dart, and XML. The Python tab is active. In the center of the workspace, there is a single 'print' block. The 'print' block is a green speech bubble shape with the word 'print' in white text. To its right is a blue square block containing the number '5' in white text.



# Blockly Code

Tiene variables (donde puedo recordar resultados)

## Blockly : Code



# Blockly Code

Ejemplo:

[Blockly](#) > [Demos](#) > Code

The screenshot shows the Blockly code editor interface. On the left is a vertical sidebar with a list of categories: Blocks, Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. At the top, there are tabs for different programming languages: JavaScript, Python, Dart, and XML. The main workspace contains three blocks connected in a sequence:

- A purple 'set' block with 'num' in the variable dropdown.
- A green 'prompt for' block with 'number' in the variable dropdown and the message 'Numero: ' in the text field.
- A green 'print' block with 'num' in the variable dropdown.

# Blockly Code

Ejemplo:

[Blockly](#) > [Demos](#) > Code

The screenshot shows the Blockly code editor interface. On the left is a sidebar with a list of categories: Blocks, Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. The main workspace is titled 'Code' and contains four blocks connected in a sequence:

- A 'set num to' block with a 'prompt for number with message' block nested inside. The message field contains the text "Numero: " with double quotes.
- A 'print num' block.
- A 'set num\_2 to' block with a 'num + 5' block nested inside.
- A 'print num\_2' block.

# Blockly Code

Ejemplo:

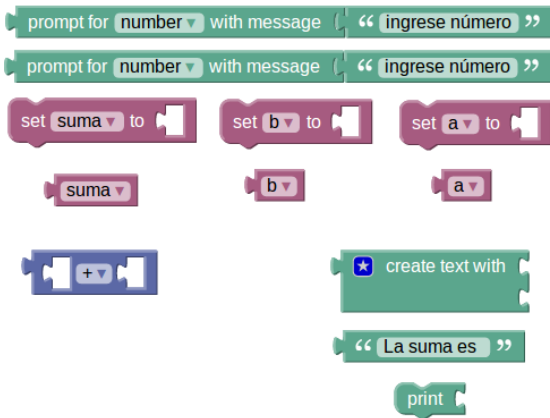
[Blockly](#) > [Demos](#) > Code

The screenshot shows the Blockly code editor interface. On the left is a sidebar with a menu of categories: Blocks, Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. The main workspace contains three blocks:

- A purple 'set' block: 'set num to prompt for number with message "Numero: "'
- A purple 'set' block: 'set num\_2 to num + 5'
- A green 'print' block: 'print create text with num= num\_2='

# Blockly Code

**Problema 1:** Cree un programa que pida dos números al usuario, los sume y luego los muestre en pantalla. Use los bloques presentes en este [link](#).



# Blockly Code

Tiene loops (while)

## Blockly : Code

Blocks	JavaScript	Python	Dart	XML
Logic				
<b>Loops</b>				
Math				
Text				
Lists				
Colour				
Variables				
Functions				

The image shows two examples of Blockly loop blocks. The first is a 'repeat' block with a dropdown menu set to '10' and the text 'times'. The second is a 'repeat' block with a dropdown menu set to 'while'.

# Blockly Code

Ejemplo:

[Blockly](#) > [Demos](#) > Code

Blocks	JavaScript	Python	Dart	XML
Logic				
Loops				
Math				
Text				
Lists				
Colour				
Variables				
Functions				

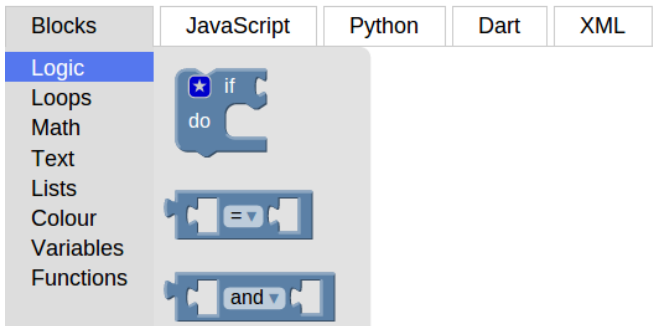
```
set num to prompt for number with message "Numero: "
repeat while num < 5
do
  print num
  change num by 1
```

The image shows a screenshot of the Blockly code editor. On the left, there is a sidebar with a list of block categories: Blocks, Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. The main workspace displays a code snippet in a Python-like syntax. The code consists of four lines: a 'set' block to prompt for a number, a 'repeat while' loop with the condition 'num < 5', and a 'do' block containing a 'print' block and a 'change num by 1' block.

# Blockly Code

Tiene control de flujo (if-else)

## Blockly : Code





# Blockly Code

Ejemplo:

[Blockly](#) > [Demos](#) > Code

Blocks	JavaScript	Python	Dart	XML
Logic				
Loops				
Math				
Text				
Lists				
Colour				
Variables				
Functions				

```
set num to prompt for number with message "Numero: "
```

```
repeat while num != 0
```

```
do
```

```
  if num < 0
```

```
  do
```

```
    set num to num + 1
```

```
  else
```

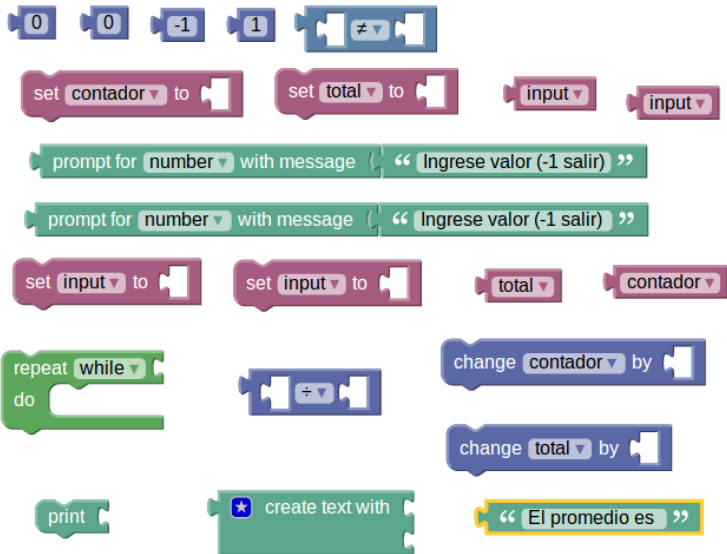
```
    set num to num - 1
```

```
print num
```

# Blockly Code

**Problema 2:** Cree un programa que calcule el promedio de los números ingresados por el usuario. Mientras no se ingrese un -1, el programa debe seguir pidiendo datos. Cuando se ingrese un -1, el programa muestra el promedio y finaliza. Use los bloques presentes en este **link**.

# Blockly Code



# Blockly Code

**Problema 3:** Programe un marcador de un partido de fútbol. El usuario debe indicar quién anotó un gol (local o visita) o si terminó el partido. Con cada anotación debe actualizarse y mostrarse el marcador. Al finalizar el partido, se debe indicar qué equipo ganó el encuentro.