

UNDERCODE

TALLER DE PYTHON



TUTOR

ANTRAX

EXTRA

EJERCITACIÓN
SUPLEMENTO EXTRA
(PYTHON EN
ANDROID)

TEMAS

INTRODUCCIÓN
INSTALACIÓN
PRIMEROS PASOS
SCRIPS
VARIABLES
CONCATENACIONES

1. Introducción a Python

En este taller veremos Python desde cero. Empezaremos por descargar nuestro intérprete e iremos avanzando en conocimientos hasta llegar a desarrollar nuestros propios programas.

Python es un lenguaje scripting Por lo tanto se interpreta en lugar de compilarse, a diferencia de otros lenguajes que es necesario pasarlos a binarios para que puedan ejecutarse

Es un lenguaje multiplataforma, ya que no solo se usa en Windows, sino que también se puede usar en Linux, MAC, etc.

Una de las ventajas es que es un lenguaje muy prolijo y entendible a la hora de programarlo y esto facilita la tarea de aprenderlo.


A lo largo de este taller, veremos lo potente e importante que puede ser saber Python.

2. Instalación

Lo que primero debemos hacer, es saber si nuestro sistema operativo es de 32 o 64 bits. Para ello vamos a:

Inicio >> Panel de control >> Sistema

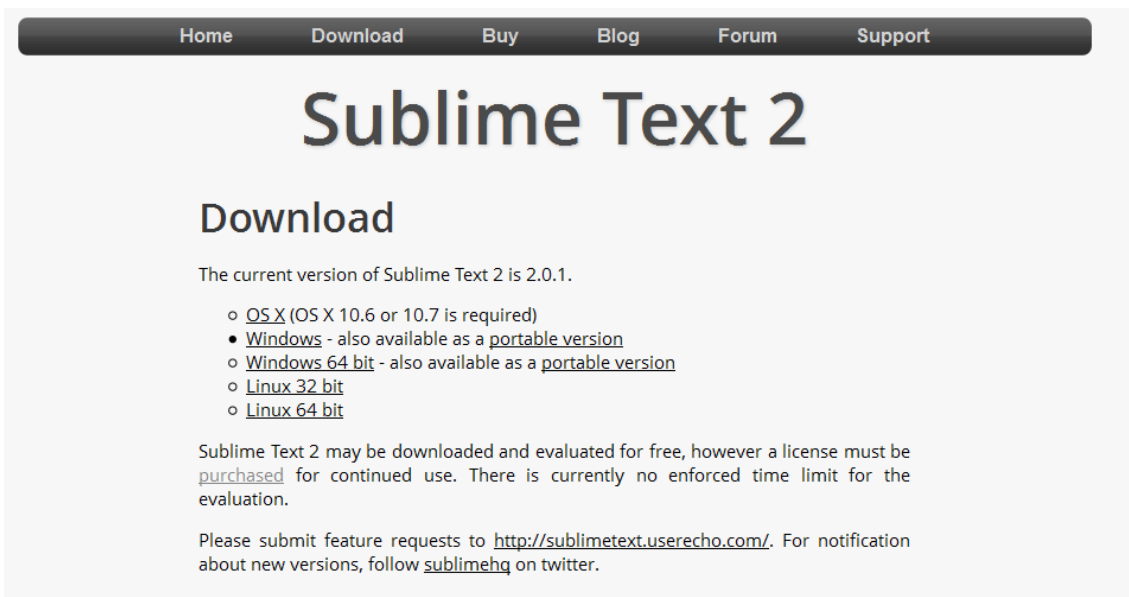
Sistema

Evaluación:	 Evaluación de la experiencia en Windows
Procesador:	Intel(R) Core(TM)2 Duo CPU T6600 @ 2.20GHz 2.20 GHz
Memoria instalada (RAM):	4,00 GB (2,98 GB utilizable)
Tipo de sistema:	Sistema operativo de 32 bits
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

En mi caso estoy utilizando un sistema operativo de 32 bits.

Lo que necesitaremos para poder programar en este lenguaje, será un editor de texto y el intérprete de Python.

Con respecto al editor de texto, pueden utilizar cualquiera que les agrade o les resulte cómodo, yo utilizare Sublime Text 2 que lo pueden descargar de acá: <http://www.sublimetext.com/2>



The screenshot shows the Sublime Text 2 website. At the top, there is a navigation bar with links for Home, Download, Buy, Blog, Forum, and Support. The main heading is "Sublime Text 2". Below it, the section is titled "Download". The text states that the current version is 2.0.1. A list of operating systems is provided: OS X (10.6 or 10.7 required), Windows (with a note about portable versions), Linux 32-bit, and Linux 64-bit. A paragraph explains that the software is free to evaluate but requires a license for continued use. Finally, it provides instructions on how to submit feature requests and follow updates on Twitter.

Y al intérprete de Python de acá <http://www.python.org/download/>

2.1. Instalación en Windows:

Start with one of these versions for learning Python or if you want the most stability; they're both considered stable production releases

If you don't know which version to use, try Python 3.3. Some existing third-party software is not yet compatible with Python 3; if you need such software, you can download Python 2.7.x instead.

For the MD5 checksums and OpenPGP signatures, look at the [detailed Python 3.3.0](#) page:

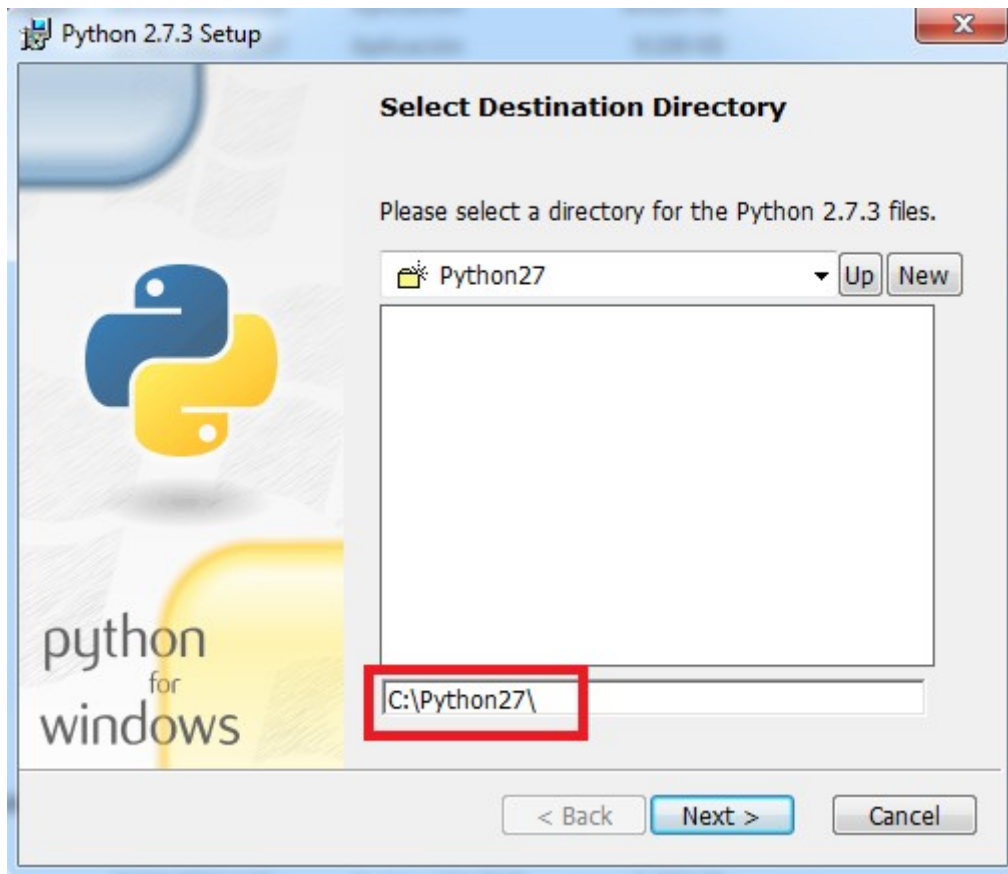
- [Python 3.3.0 Windows x86 MSI Installer](#) (Windows binary -- does not include source)
- [Python 3.3.0 Windows X86-64 MSI Installer](#) (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- [Python 3.3.0 Mac OS X 64-bit/32-bit x86-64/i386 Installer](#) (for Mac OS X 10.6 and later [2])
- [Python 3.3.0 Mac OS X 32-bit i386/PPC Installer](#) (for Mac OS X 10.5 and later [2])
- [Python 3.3.0 compressed source tarball](#) (for Linux, Unix or Mac OS X)
- [Python 3.3.0 bziped source tarball](#) (for Linux, Unix or Mac OS X, more compressed)

For the MD5 checksums and OpenPGP signatures, look at the [detailed Python 2.7.3](#) page:

- [Python 2.7.3 Windows Installer](#) (Windows binary -- does not include source)
- [Python 2.7.3 Windows X86-64 Installer](#) (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- [Python 2.7.3 Mac OS X 64-bit/32-bit x86-64/i386 Installer](#) (for Mac OS X 10.6 and later [2])
- [Python 2.7.3 Mac OS X 32-bit i386/PPC Installer](#) (for Mac OS X 10.3 through 10.6 [2])
- [Python 2.7.3 compressed source tarball](#) (for Linux, Unix or Mac OS X)
- [Python 2.7.3 bziped source tarball](#) (for Linux, Unix or Mac OS X, more compressed)

NOTA: Usaremos por ahora Python 2.x debido a que es la versión estable. Más adelante, veremos algo de Python 3.x

Una vez descargado, ejecutamos el MSI que descargamos



Es muy importante saber la ruta en donde se instalara. Yo les recomiendo dejarla en donde viene por defecto que es en **C:\Python27**

Damos en Next y continuamos la instalación hasta que finalice

Ahora abrimos una consola:

INICIO > EJECUTAR > COMMAND (En Windows 95,98)

INICIO > EJECUTAR > CMD (En Windows XP)

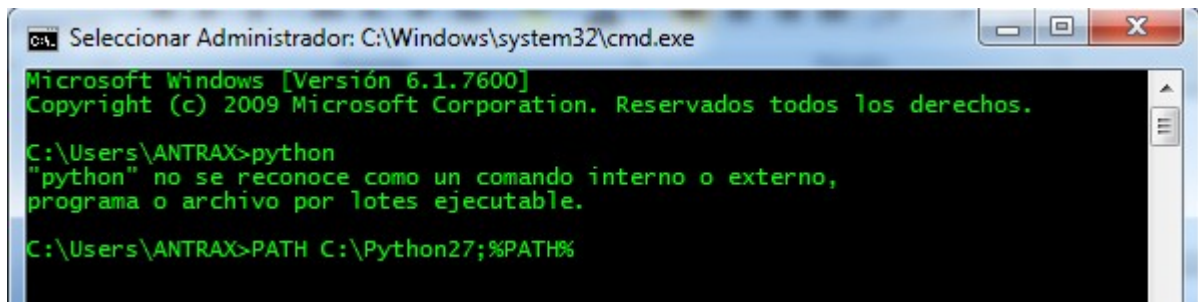
INICIO > EN LA BARRA BUSCAR > CMD (En Windows Vista y 7)

Y dentro de la consola escribimos **python**, lo más probable, es que les aparezca el siguiente error:

"python" no se reconoce como un comando interno o externo, programa o archivo por lotes ejecutable.

Esto se debe a que no está añadido en las variables de entorno, asique tipeamos lo siguiente:

PATH C:\Python27;%PATH%

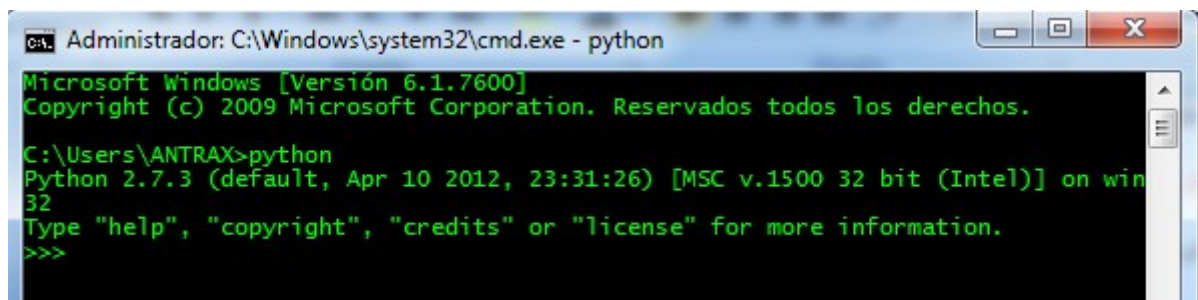


```
Seleccinar Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\ANTRAX>python
"python" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ANTRAX>PATH C:\Python27;%PATH%
```

Presionamos enter, cerramos la consola y la volvemos a abrir... Ahora si tipeamos python, debería aparecernos algo como esto:



```
Administrador: C:\Windows\system32\cmd.exe - python
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\ANTRAX>python
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Esto nos indica que lo hemos hecho bien, y ya podemos comenzar a programar.

2.2. Instalación en Linux

La instalación en Linux es mucho más fácil... abrimos una consola (CTRL + ALT + T) y tipeamos:

Sudo apt-get install python

Y esto nos instalara Python en Linux.

Luego al comienzo de cada script, debemos añadir la línea:

#!/usr/bin/pyhton27

Para que el intérprete busque ahí las librerías.

3. Primeros pasos en Python

Comenzaremos con programas básicos. Como dijimos en la introducción, Python es un lenguaje scripting ya que no es necesario compilarlo para poder ver el programa en funcionamiento.

Dentro de la consola, pondremos **5+7** y veremos lo que pasa

```

Administrador: C:\Windows\system32\cmd.exe - python
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\ANTRAX>python
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> 5+7
12
>>>

```

Rápidamente nos hará el cálculo.

Python se caracteriza por ser una potente calculadora. Es por eso que ahora probaremos con algo más avanzado.

En la consola pondremos **56+28-34*6**

```

>>> 56+28-34*6
-120
>>>

```

Ahora también podemos trabajar con variables, asignándoles algún valor.

Podemos probar con **X=7**

Seguido a esto colocamos **105*X**

```

>>> X=7
>>> 105*X
735

```

Hasta el momento se ve muy fácil y básico, aunque también muy tedioso ya que tipear comandos desde la consola no es muy gustoso, es por eso que ahora armaremos nuestro primer script.

4. Operadores Aritméticos

A continuación veremos los operadores que se suelen utilizar con frecuencia en Python

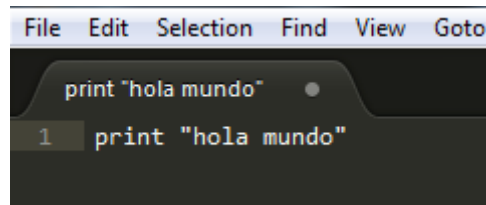
Operación	Operador	Aridad	Asociatividad	Prioridad
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	-----	2
Cambio de Signo	-	Unario	-----	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo(resto)	%	Binario	Por la izquierda	
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4

5. Nuestro primer Script

Abrimos nuestro editor de texto y haremos el famoso “Hola Mundo”

Dentro del editor pondremos:

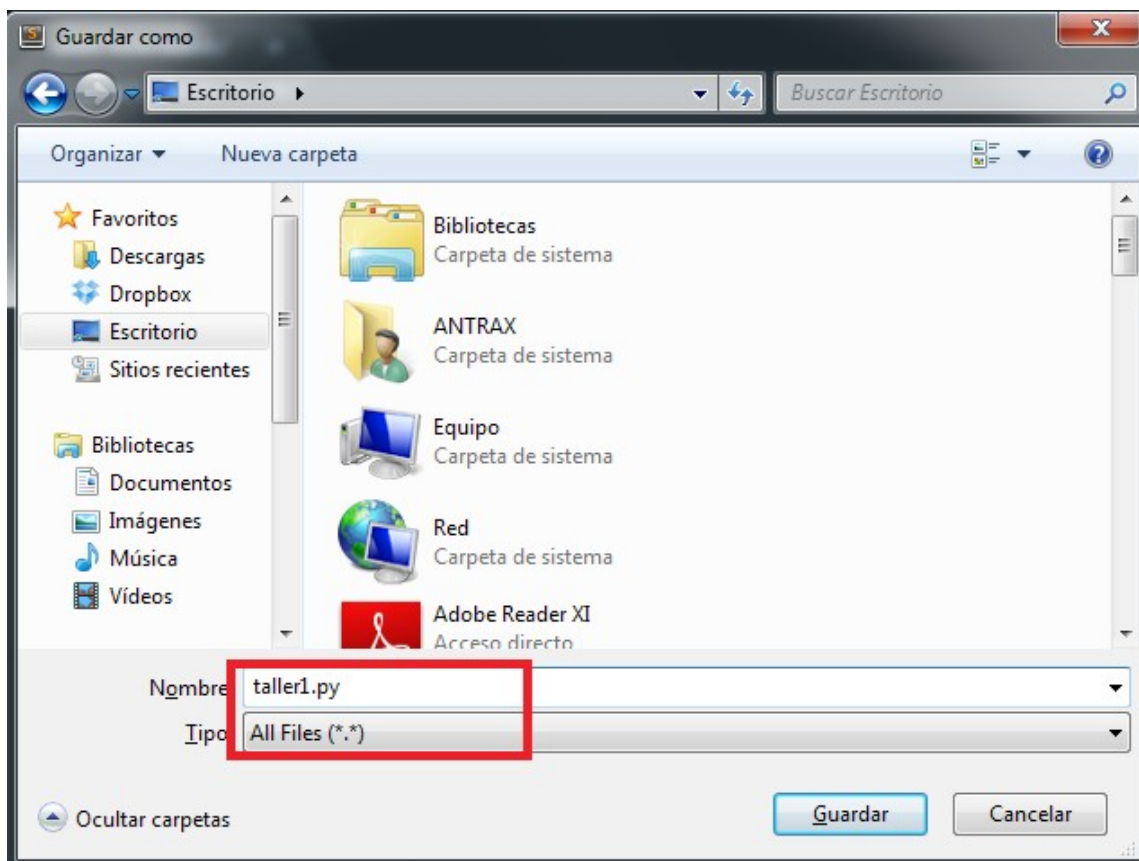
```
print "Hola Mundo"
```



```
File Edit Selection Find View Goto  
print "hola mundo"  
1 print "hola mundo"
```

La sintaxis print sirve para mostrar una cadena de valores en pantalla. Un poco más adelante lo veremos más en detalle.

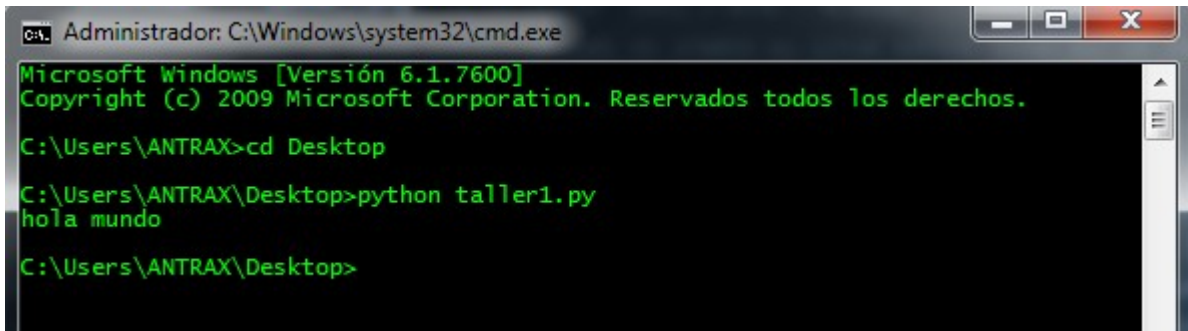
Ahora lo guardamos (CTRL + S) o (File >>Save As..), lo guardamos con extensión *.py y nos aseguramos que en tipo de archivo diga “All Files (*.*)”



Y ahora lo correremos nuestro script desde la consola.

Es importante saber en donde guardamos nuestro script, yo lo hice en el escritorio, así que desde la consola, debemos pararnos en el directorio en donde está guardado y tipeamos

```
python taller1.py
```

A screenshot of a Windows command prompt window titled "Administrador: C:\Windows\system32\cmd.exe". The window shows the following text:

```
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\ANTRAX>cd Desktop

C:\Users\ANTRAX\Desktop>python taller1.py
hola mundo

C:\Users\ANTRAX\Desktop>
```

Y nos muestra nuestro Hola mundo...

6. Variables

Las variables en Python pueden estar formadas por letras o números, pero siempre deben comenzar con letras o un guion bajo. Algo a tener en cuenta, es que este lenguaje es sensible a mayúsculas y minúsculas. Es decir, para el intérprete no es lo mismo "A" que "a".

Las variables en Python pueden ser de tipo:

Numéricas:

Enteros (Ej. 7) Son números positivos y negativos, que no poseen decimales.

Dentro de los enteros, tenemos dos tipos, Cortos (int) o Largos (long) expresados de la siguiente forma:

Entero_corto = 7

Entero_largo = 7L

Flotante (Ej. 7,9) Son todos números reales que posean decimales y se representan como (float)

Flotante = 1.345

Compleja o imaginaria (Ej. 2+3i) Son aquellos que poseen una parte real y otra imaginaria y se representan como (complex)

Imag = 4.3 + 7i

Cadenas de Texto: (Ej.: "ANTRAX"), las cadenas de texto es simplemente texto y pueden estar entre comillas dobles o simples

Ej1 = "ANTRAX"

Ej2 = 'ANTRAX'

Booleano: True, False (Verdadero o Falso), Estos dos son muy utilizados en bucles o condiciones.

7. Concatenaciones

Como vimos antes, es posible sumar, restar, multiplicar, dividir, etc.... Pero no solo se hace con números sino que también con letras, palabras y variables. Realizaremos el siguiente ejemplo

```
File Edit Selection Find View Goto Tools
taller1.py
1 x = "underc0de"
2 print "Hola " + x
```

Como resultado tendremos

```
C:\Users\ANTRAX\Desktop>python taller1.py
Hola underc0de
```

Ya que sumo las dos cadenas de texto, la del Hola y X que tenía almacenado otra cadena de texto. Otra forma de concatenar dos cadenas de texto sería:

Print "Hola" + "underc0de"

Y con eso obtendremos el mismo resultado que la forma anterior.

Ahora veremos un ejemplo un poco más complejo, en donde concatenaremos números, variables y cadenas de texto

```
File Edit Selection Find View Goto Tools Project Preferences Help
taller1.py x
1 a = "ANTRAX"
2 print "hola " + a + " el resultado de 145 + 34 es ",145+34
```

Explico rápidamente, guarde mi Nick en la variable a, luego en el print, lo que hice fue mostrar la cadena de texto "hola" concatenado con la variable "a" que es mi Nick, hasta aquí tenemos "hola ANTRAX" luego lo concatene con "el resultado de 145 + 34 es" y hasta aquí tendríamos "hola ANTRAX el resultado de 145 + 34 es" finalmente para concatenar con números, es necesario colocar una coma. Aquí lo que hace es resolver la suma algebraica de 145+34 y como resultado final tenemos:

```
C:\Users\ANTRAX\Desktop>python taller1.py
hola ANTRAX la suma de 145+34 es 179
```

8. Comentarios

Los comentarios sirven para ayudar a entender el código, para explicar una función o simplemente agregar algún texto.

Es importante saber que estos comentarios no son ejecutados a la hora de correr el programa. Para colocar un comentario, solo basta con poner el carácter numeral, seguido del comentario. Ejemplo:

#esto es un comentario

Aquí lo mostrare en el editor de texto:

```
1 # Hola Mundo
2 print "Hola Mundo" # Esto mostrara un hola mundo en pantalla.
3 a = 13
4 b = 98
5 c = a + b #suma de variables
6 print c #muestra el resultado
7
```

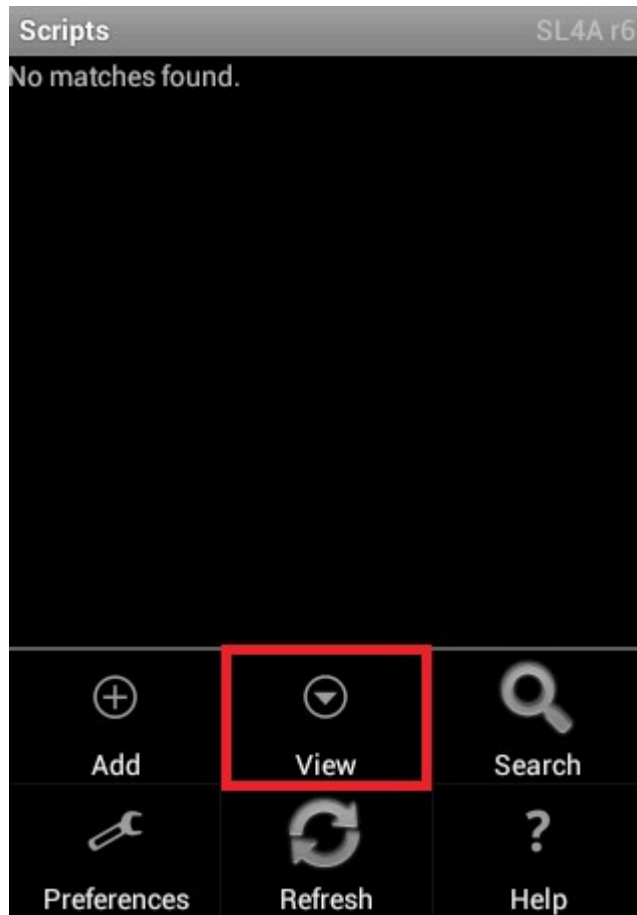
Los comentarios suelen ser útiles a la hora de hacer público un script, de esta forma, la persona que lo lea pueda entender mejor que hace cada función del código.

9. Python desde Android

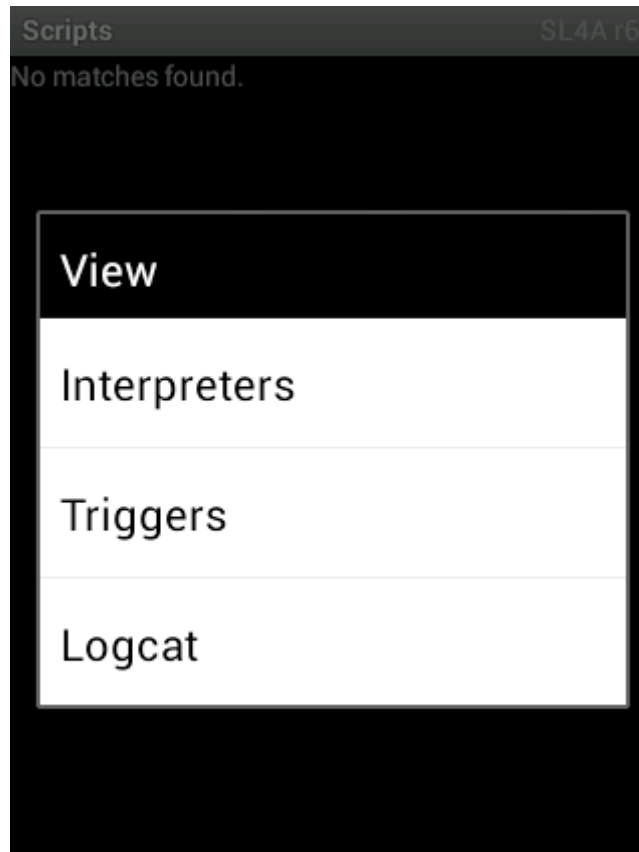
Como parte extra de este taller, les enseñare a instalar Python en algún dispositivo móvil con Android, para que puedan programar desde el.

Lo que necesitamos es entrar a esta página: <http://code.google.com/p/android-scripting/downloads/list> y descargar el último APK del SL4a. Actualmente la última versión es la r6.

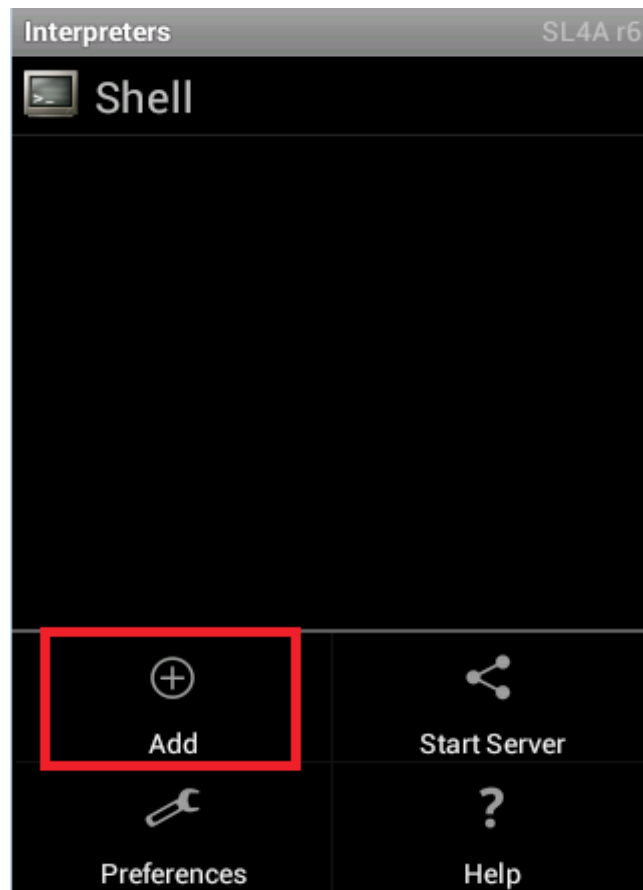
Instalamos como cualquier apk de android y lo abrimos.



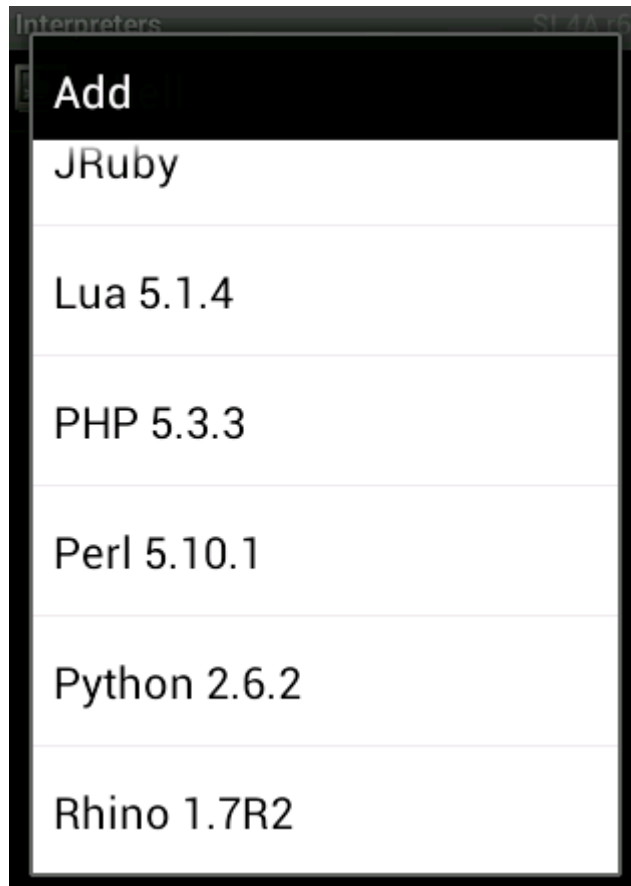
Presionamos en View



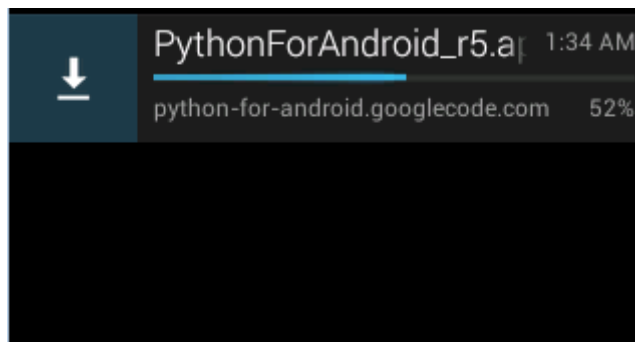
Entramos a Intérpretes



Y agregamos a Python... A demás de python, como podrán ver, pueden agregar Perl, Ruby, PHP entre otros



Al presionar sobre Python, lo comienza a descargar



Luego lo instalamos



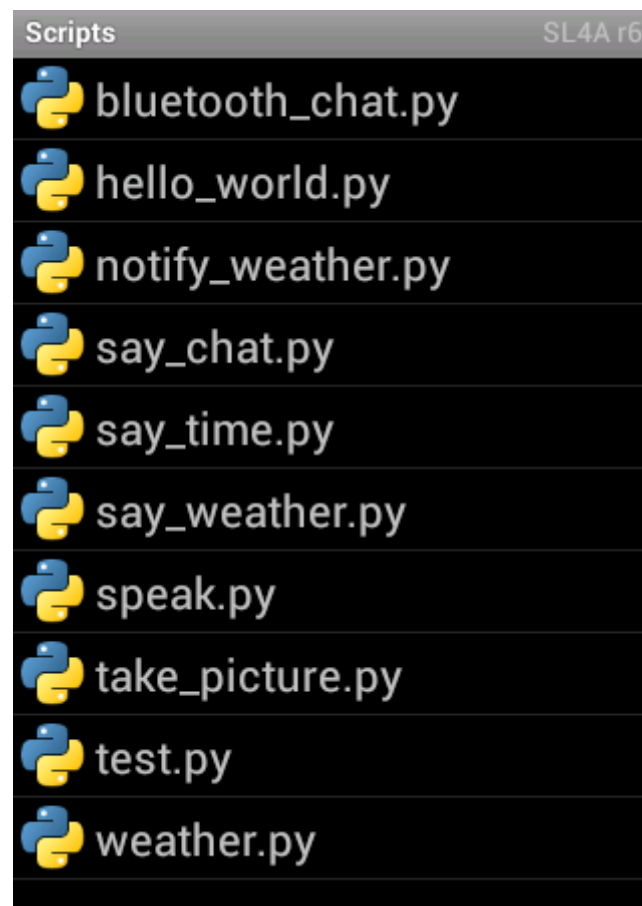
Esto suele demorar algunos minutos en descargar los módulos y librerías



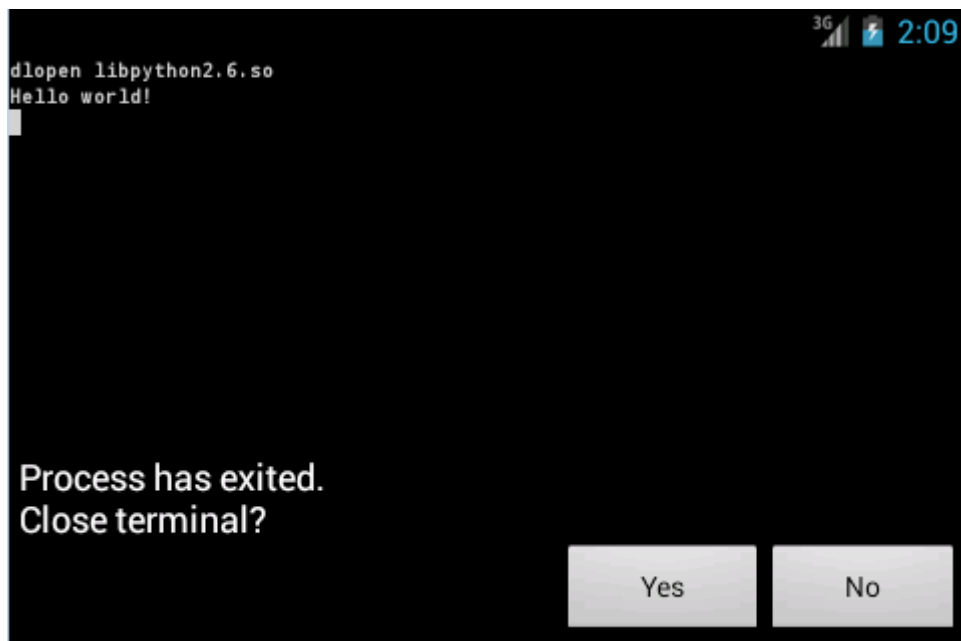
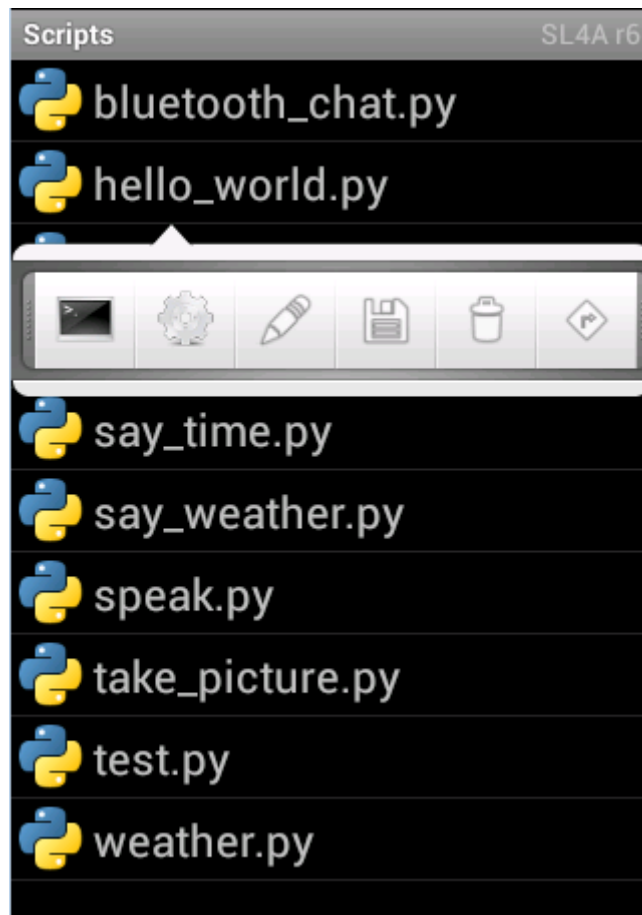
Una vez que termine, veremos algo como esto



Volvemos a abrir SL4a y podremos ver scripts de muestra hechos en Python



Finalmente clickeando sobre alguno de ellos, tendremos la opción de editarlos o ejecutarlos.



10. Ejercicios

1.- Realizar un código que concatene 3 variables, de las cuales 2 sean numéricas.

El usuario X lleva Y días y Z horas conectado

2.- Realizar un código para cada uno de los siguientes ejercicios. Cada número debe ser una variable

a. $273 * 4 + 97 * 5 - 1200 / 6$

b. $1470 / 5 + 400 * 2 - 49**2$

c. $15**5 / 50 - 37 * 4$

Todos los resultados deben mostrarse de la siguiente forma:

El resultado es: