

Programación en Python I



Ing. Wilmer Garzón Alfonso

wilmer.garzon@escuelaing.edu.co

www.wilmergarzon.com.co



Código de Honor



ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

Como miembro de la comunidad académica de la Escuela Colombiana de Ingeniería, los valores éticos y la integridad son tan importantes como la excelencia académica. En este curso se espera que los estudiantes adopten el siguiente código de honor:

"Me comprometo a seguir los más altos estándares de integridad académica."

Integridad académica se refiere al opuesto de plagiar, engañar, 'hacer trampa', etc.

Por eso es importante ser honesto, dar crédito a quien lo merece y respetar el trabajo de los demás. En particular, el acto de entregar un programa de computador ajeno como propio constituye un acto de plagio; cambiar el nombre de las variables, agregar o eliminar comentarios y reorganizar comandos no cambia el hecho de que se está copiando el programa de alguien más.

Para más detalles consultar el Reglamento de Estudiantes, partes 2 y 4.^[1]

^[1] <http://www.camilorocha.info/>



Agenda

- Historia
- ¿Qué es Python?
- Características
- Ventajas - Desventajas
- Metodología
- Configuración
- Scripts - Ejecución
- Variables - Identificadores
- Entrada y Salida de datos
- Referencias



Historia

- Python fue creado por el científico computacional **Guido van Rossum**, Holandés de 58 años a comienzos de los 90s.
- Trabajó para Google, Dropbox, entre otras.
- El nombre del lenguaje está inspirado por el popular grupo cómico británico Monty Python.



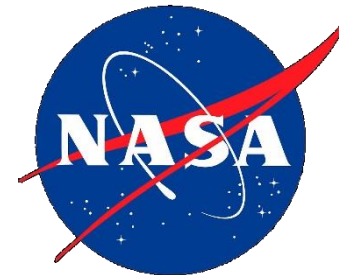
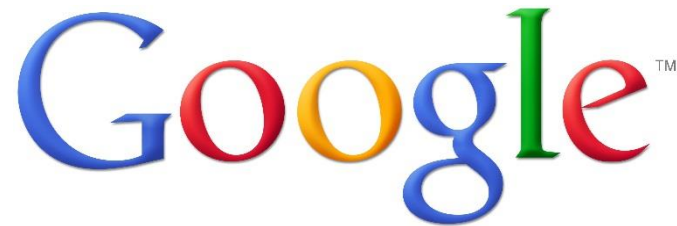
¿Qué es Python?

- Un lenguaje de programación de fuente abierta.
- Imperativo (existe la noción de estado que se modifica por medio de asignaciones).
- Extensible usando C o Java.



¿Qué es Python?

- Un lenguaje de programación dinámico y orientado a objetos.
- El principal objetivo es la facilidad, para aprenderlo, para diseñar y programar.
- Grandes aplicaciones están diseñadas en Python:





Características

- Imperativo.
- Interpretado (el código se interpreta a medida que se va haciendo parsing, NO existe un proceso de compilación).
- Minimalista, no es necesario: ';' '{ }' '\n'
- Muy denso: poco código hace mucho.



Características

- Diferentes opciones para organizar código: funciones, clases, módulos, y paquetes.
- Soporta objetos y estructuras de datos de alto nivel: strings, listas, diccionarios, etc.
- Se puede hacer en Python todo lo que se hace con C# o Java, o más.



Características

- Cientos de librerías que permiten satisfacer todo tipo de necesidades.
- No es necesario declarar constantes y variables antes de utilizarlas.
- La primera vez que se ejecuta un script de Python se compila y genera bytecode que es luego interpretado.
- Alta velocidad de desarrollo y buen rendimiento.



Ventajas

- Rapidez para desarrollar
- Sencillez y velocidad
- Altamente portable
- “Toda expresión” es un programa, Ej.: $3 + 4$
- Cuenta con cientos o miles de librerías
- Soporta varias bases de datos



Desventajas

- Más lento en comparación con lenguajes de programación como C o Java, los cuales son compilados. Esto se debe, principalmente, al hecho de ser interpretado.
- Algunas versiones no son compatibles entre versiones.
- Incompatibilidad de librerías entre la versión 2.x y 3.x



Metodología

- Estudiantes Proactivos (aprender haciendo)
- Internet, libros, videos, tutoriales.... serán de gran utilidad relacionados con el lenguaje de programación Python.
- Resolveremos problemas de programación competitiva. Aumentando la complejidad de los problemas día a día:
 - <http://uva.onlinejudge.org/>
 - <https://www.hackerrank.com/>
 - <https://www.topcoder.com/>
 - Etc....



Configuración (Tarea 0)

- Python está disponible para Windows, Mac o Linux.
- Última versión es la 3.5
- Hay varios entornos de programación para Python, algunos de ellos:
 - PyCharm
 - Enthought Canopy Python
 - GNU Emacs
 - IDLE

<https://www.python.org/>



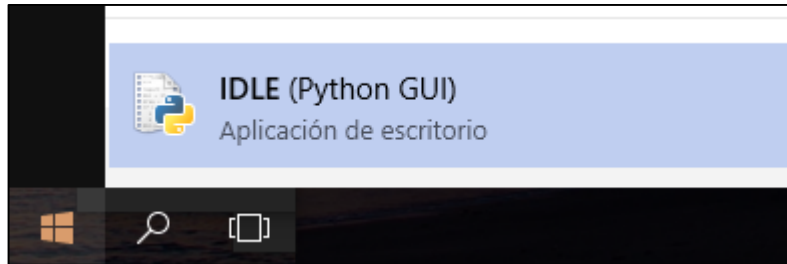
Programas en Python

- También llamados **Script**
- Un **script** es una secuencia de **definiciones** y **comandos**
 - Las definiciones son evaluadas y los comandos son ejecutados por el interprete de Python en la consola
 - Los comandos le indica al interprete que tiene que hacer.
- Los scripts se almacenan en un archivo con extensión **.py**



Ejecución de Programas

1. Desde el interprete de Python:

A screenshot of a "Python 3.4.1 Shell" window. The window has a title bar with the Python logo and standard window controls. The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area displays the following information: "Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32", followed by "Type 'copyright', 'credits' or 'license()' for more information.", and then the prompt ">>>" on a new line. A cursor is visible on the line following the prompt, indicating the next input line.

```
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.
>>>
>>> |
```



Ejecución de Programas

2. Llamando al archivo .py desde el sistema operativo

A screenshot of a Python IDE window. The title bar reads "Python 3.4.1: prueba.py - C:/Python34/prueba.py". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code editor contains the line `print("Hello World!!!")` in a syntax-highlighted font.

```
Python 3.4.1: prueba.py - C:/Python34/prueba.py
File Edit Format Run Options Windows Help
print("Hello World!!!")
```

```
C:\Python34>python prueba.py
Hello World!!!
```




Variables

- Son lugares reservados en memoria donde se puede almacenar datos.
- Toda variable tiene Nombre, Tipo y Contenido.
 - *Nombre: Alfanumerico*
 - *Tipo: Str, Int, Float, Complex, Bool,....*
 - *Contenido: Relacionado con el tipo de la variable*
- Python automáticamente reserva el espacio en memoria para cada variable.



Tipos de datos

➤ Numéricos:

- int
- float
- complex

```
>>> x = 8
>>> int(x)
8
>>> float(x)
8.0
>>> complex(4, 0.2)
(4+0.2j)
>>> type(x)
<class 'int'>
```

➤ Booleanos

- bool

```
>>> a = True
>>> False != True
True
>>> True != True
False
```



Identificadores

- Permiten distinguir variables, módulos y funciones.
 - Deben comenzar con una letra
 - Python es case sensitive
- Variable es diferente a un identificador. El nombre de la variable es un identificador, pero la variable tiene propiedades
- No se permiten palabras reservadas: *and, global, or, assert, else, if, pass, break, except, import, print, class, exec, in, raise, continue, finally, is, return, def, for, lambda, try, del, from, not, while, int, float, bool*



Inicialización de variables

- En Python no es necesario crear variables.
- El signo “=” es utilizado para asignarle un valor a una variable.
 - `x = 2`
 - `a = “hola”`
 - `x = x+1`

**Obtiene el valor asignado a la variable x
Al valor obtenido le adiciona 1
El valor resultado lo vuelve a dejar en la variable x**



Operadores

➤ Aritméticos:

() + - * / % ** //

➤ Relacionales:

> >= < <= == !=

➤ Lógicos:

and or not



Entrada y Salida

- Existen diversas formas de leer datos y generar salida a partir de estos. En este curso trabajaremos dos formas:
 - 1. Lectura de información por medio del teclado.**
 - 2. Lectura de información desde archivos planos.**



Entrada y Salida

1. Lectura de información por medio del teclado.

Para ingresar información al programa `Input(" ")`

```
>>> name = input('Enter your name:')  
Enter your name:Wilmer Garzón
```

➤ Para mostrar la información `Print(" ")`

```
name = input("Tu nombre es:")  
a = int(input("Primer Número"))  
b = int(input("Segundo Número"))  
s = a+b  
print("Hola ", name, "!!!")  
print("La suma es",s)
```

```
>>>  
Tu nombre es:Pedro  
Primer Número25  
Segundo Número2  
Hola  Pedro !!!  
La suma es 27
```

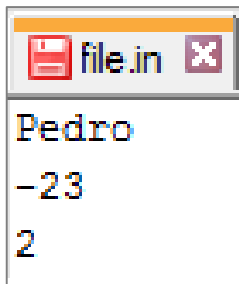


Entrada y Salida

2. Lectura de información desde archivos planos.

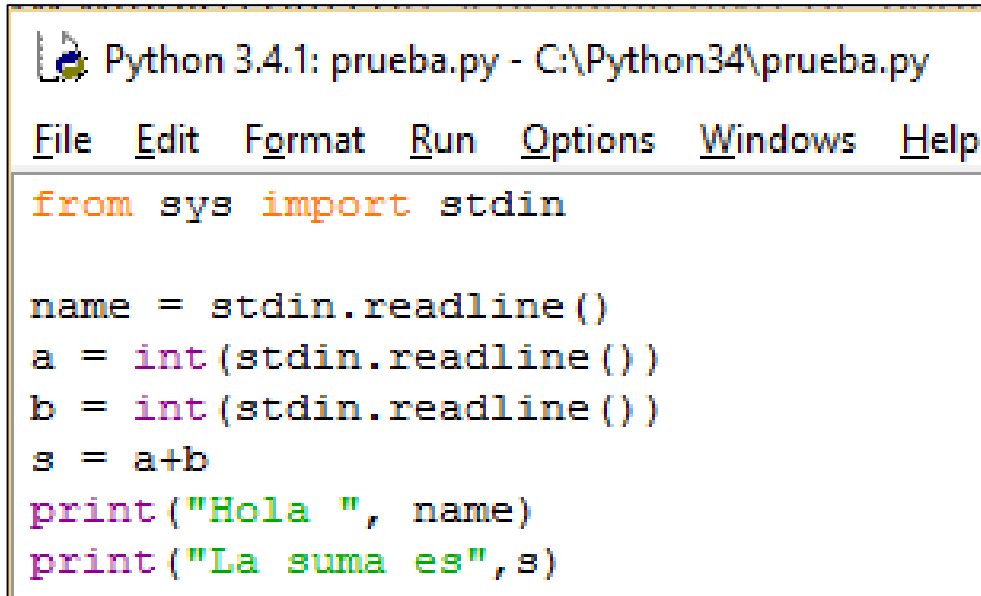
- Los problemas de la arena se resuelven por medio de la salida y entrada estándar. Se utiliza la librería Sys de Python

Datos



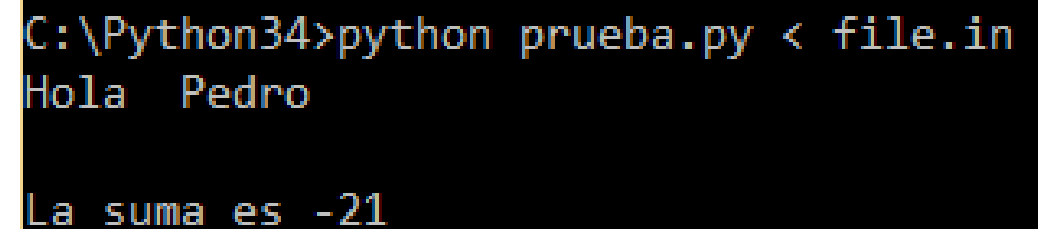
```
file.in
Pedro
-23
2
```

File.in



```
Python 3.4.1: prueba.py - C:\Python34\prueba.py
File Edit Format Run Options Windows Help
from sys import stdin

name = stdin.readline()
a = int(stdin.readline())
b = int(stdin.readline())
s = a+b
print("Hola ", name)
print("La suma es",s)
```



```
C:\Python34>python prueba.py < file.in
Hola Pedro

La suma es -21
```



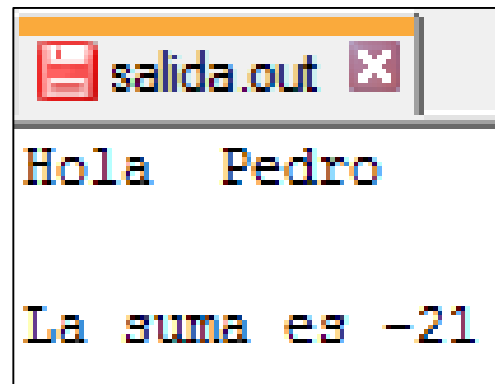

Entrada y Salida

2. Lectura de información desde archivos planos.

```
C:\Python34>python prueba.py < file.in > salida.out
```

Archivo Python Archivo a Leer Archivo Salida

Archivo de Salida



salida.out

```
Hola  Pedro

La suma es -21
```



Referencias

- Introduction to Computation and Programming Using Python, revised and expanded edition, John V Guttag, MIT Press.
- Python Programming: An Introduction to Computer Science, John Zelle.
- MITx's Introduction to Computer Science and Programming Using Python.
- <http://paginaspersonales.deusto.es/dipina/teaching.html>
- <https://courses.edx.org/courses/course-v1:UTAx+CSE1309x+2016T1/>
- <https://www.python.org/>
- <http://www.camilorocha.info/>