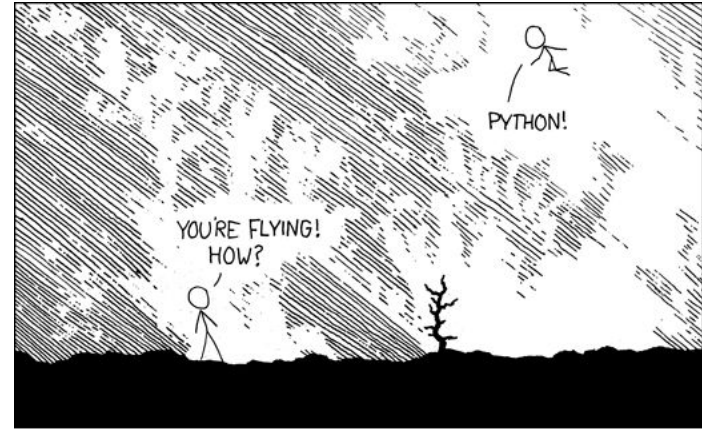




CUSOL (Comunidad Universitaria de
Software Libre)

Características

- Lenguaje de **alto nivel**.
- Multiparadigma
 - ◆ POO
 - ◆ Programación Funcional
- Multiplataforma
- Dinámicamente tipado
- Interpretado
- Open source :)



```
>>> import antigravity
```

¿Interpretado o Compilado?



Figure 1.1: An interpreter processes the program a little at a time, alternately reading lines and performing computations.

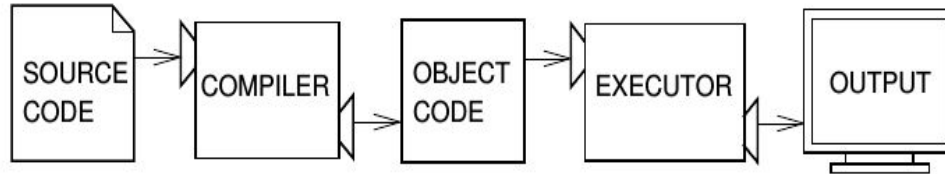


Figure 1.2: A compiler translates source code into object code, which is run by a hardware executor.

Just in time Compilation: Compilacion en tiempo real empieza a cerrar la brecha.

Ejercicio 1

Haz tu primer programa en python. Ejecuta en el intérprete

```
>>print "Hello world"
```

```
>>1+1
```

Haz ahora un script con las mismas instrucciones, usa el editor que quieras.
Ejs: Spyder, Notebook, Geany, etc. Ejecuta el archivo.

Un poco de Historia

Presentado en 1991 por **Guido Van Rossum (BDFL)**, en honor a Monty Python.

Liberado con Python Software Foundation License, compatible con la Licencia pública general de GNU a partir de la versión 2.1.1

¡Es de todos y para todos!

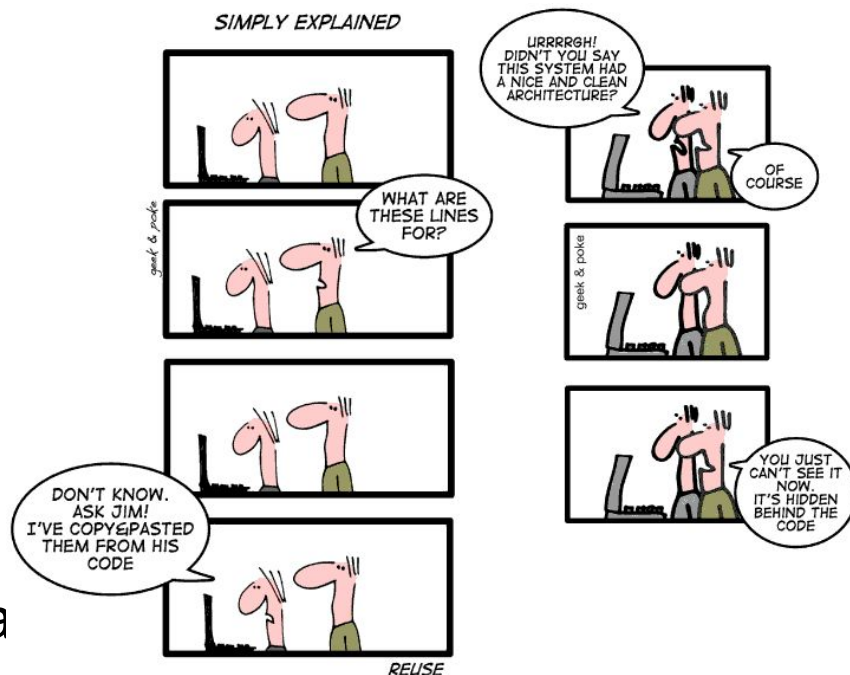


...Y filosofía

La filosofía de python es:

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo..

Python enfatiza la importancia de la productividad y legibilidad del código.



```
>>> import this
```

Usan Python...



Google



Instagram



<https://wiki.python.org/moin/OrganizationsUsingPython>

¿Para qué sirve?

- Scripting.
- Programación web (Django, Flask).
- Cálculo Científico.
- Data science, machine learning.
- Aplicaciones stand-alone.

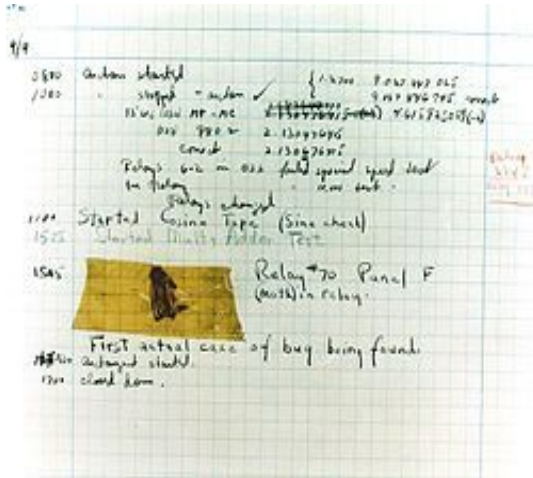
¿Que es un programa?

Es una secuencia de instrucciones.

- Entrada
- Salida
- Matemática
- Condiciones
- Repeticiones

Debugging

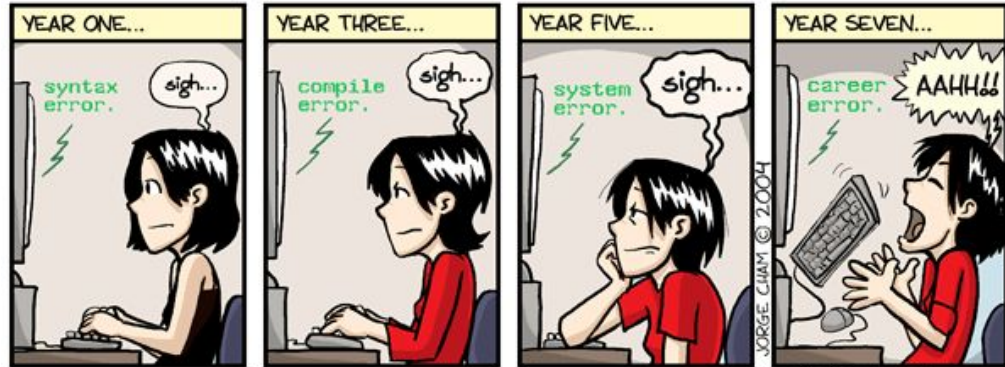
Los programadores son humanos...



El termino "bug" fue usado por la pionera en computación [Grace Hopper](#).

Encontrar errores puede ser frustrante, pero también interesante :)

RESIGNATION: THE EVOLUTION OF THE SIGH



Debugging: Tipos de error

- Sintaxis

```
>>> print "Hello world
      File "<stdin>", line 1
        print "Hello world
              ^
SyntaxError: EOL while scanning string literal
>>> █
```

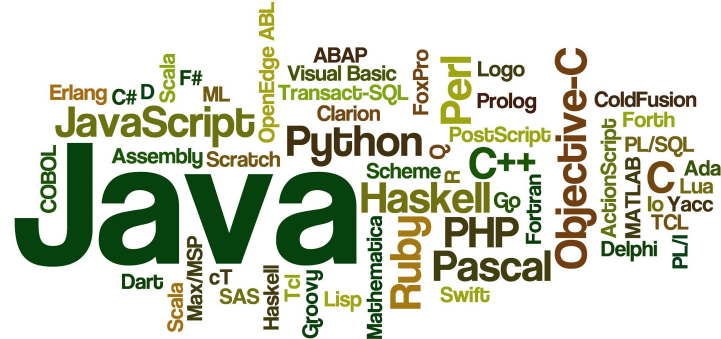
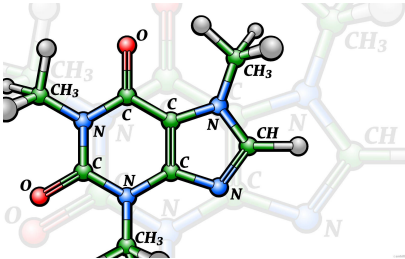
- Ejecución
- Semántica... “El programa que escribiste no es el que querías.”

Lenguajes formales y Naturales

- Naturales



- Formales



Ejercicios

1. Visitar <https://www.python.org/> → Info, documentacion. Hay que LEER.
2. Abrir el intérprete de python. Digite `help()` y explore la ayuda en línea de python para algunas de sus funciones. Ejemplo:: `help('print')`
3. Usar el intérprete como calculadora con los operadores `+`, `-`, `*`, `/` ...

¿Porque usar Python?

- Filosofía
- Legibilidad
- Flexible
- Muchas librerías
- Fácil de aprender
- Rápido
- Comunidad
- Popularidad
- LIBRE!!!!

Valores y Tipos

Valor → tiene un tipo, se le asigna a una variable.

Variable → Una variable refiere a un valor.



n = 17

pi = 3.1416

mensaje = "Hola"

ASIGNACIONES

Valores y Tipos

En python se crea y se asigna al mismo tiempo las variables. **Su tipado es dinámico.**

Algunas consideraciones

1. Los nombres de variables pueden ser muy largos. Pero no se recomienda. Recuerda elegir bien los nombres de las variables para que tu código sea legible y lo entiendas.
2. Deben comenzar con una letra. (minúscula)
3. Números y letras son combinables. Teniendo en cuenta 2.
4. OJO con las Keywords.

Keywords (Palabras reservadas)

Python 2 has 31 keywords:

<code>and</code>	<code>del</code>	<code>from</code>	<code>not</code>	<code>while</code>
<code>as</code>	<code>elif</code>	<code>global</code>	<code>or</code>	<code>with</code>
<code>assert</code>	<code>else</code>	<code>if</code>	<code>pass</code>	<code>yield</code>
<code>break</code>	<code>except</code>	<code>import</code>	<code>print</code>	
<code>class</code>	<code>exec</code>	<code>in</code>	<code>raise</code>	
<code>continue</code>	<code>finally</code>	<code>is</code>	<code>return</code>	
<code>def</code>	<code>for</code>	<code>lambda</code>	<code>try</code>	

In Python 3, `exec` is no longer a keyword, but `nonlocal` is.

Tipos de Datos y Operadores

Entero: a=2

Float: b=2.0

Cadenas: s="Hola Mundo!"

Boolean: vacio=False

Listas: lista=[12, "Camila", False]

Tuplas: (1, 2, "w", ":)")

Diccionarios:

dias = {"Lunes": 1, "Martes": 2}

Lógicos

and, or, not

Aritméticos

+, -, *, /, //, %, **

De comparación

==, <, >, <=, >=, !=

Es dinámicamente tipado

Ejercicio

Abrir el intérprete de Python, crear diferentes variables.

Usar la función `type()` para imprimir el tipo de la variable.

Ej:

```
>> nombre = "Camilo"
```

```
>> type(nombre)
```

Usar los operadores `+` y `*` en cadenas, observe la operación que realiza sobre estas.

Estructuras

- De decisión

```
if condicion1:
    instruccion1
elif condicion2:
    instruccion2
...
elif condicionN:
    instruccionN
else:
    instruccion
```

Ejemplo:

```
if x > 0:
    print "Positivo"
elif x < 0:
    print "Negativo"
else:
    print "Zero"
```

Estructuras

- **For**

```
lista = [2, 3, 1, 4, 67, 21, 42, 3]
```

```
for x in lista:  
    print x
```

```
a = "Hello world!"
```

```
for x in a:  
    print a
```

- **While**

```
while condición:  
    instrucciones
```

```
contador = 0  
while (contador < 9):  
    print("El contador es: ",  
contador)  
    contador = contador + 1
```

Algunas funciones básicas

`print()`

`input()`

`f = open("archivo.txt", opcion)`

opciones: `w`, `r`, `a`, `r+`

`f.read()`

`f.write()`

`list.append()`

`len()`

`string.format()`

`type()`

`int()`

`float()`

`str()`

Scripting en línea



codingground
SIMPLY EASY CODING



codecademy



SKULPT

http://www.tutorialspoint.com/execute_python3_online.php

Python 3

“Python 2.x es obsoleto, Python 3.x es el presente y el futuro del lenguaje”

- Print
- División de enteros
- Cadenas unicode
- Iteradores



wiki.python.org/moin/Python2orPython3

<https://asmeurer.github.io/python3-presentation/slides.htm>

¿Dónde aprender más en la web?

Python 2.x o 3.x

- <https://www.coursera.org/specializations/python> **Especialización para crear plataforma para manejar, procesar y visualizar datos**
- <https://www.coursera.org/course/interactivepython1> **Programación de pequeños videojuegos para resolver problemas**
- <https://www.python.org/> **Documentación**
- <http://www.codecademy.com> **Curso interactivo**
- <https://www.edx.org/course/introduction-python-data-science-microsoft-dat208x-0> **Curso introductorio para computer science**
- **Stanford: Introducción a python científico**

Libros

- Think python. Allen B. Downey.
- Python para todos por Raúl González Duque
- Learning Python (Powerful Object-Oriented Programming). Mark Lutzç
- Muchos más...

Programando como un Pythonista!

- PEP8 python.org/dev/peps/pep-0008
- autopep8, yapf
- Zen of Python python.org/dev/peps/pep-0020/
- The Hitchhiker's Guide to Python! docs.python-guide.org/en/latest/
- Testing!!! docs.python.org/3.6/library/unittest.html
- Pip, virtualenv, pyvenv
- Notebooks Jupyter!

¡Practica y programa!