Introducción a Python

Christopher Flores Jara

Rancagua, 07/09/2023

Contenidos

1. Breve descripción

2. Introducción a Python
   Sintaxis, tipos de datos, operadores,
   estructuras de control, funciones

3. Introducción a la POO
   Clases y objetos

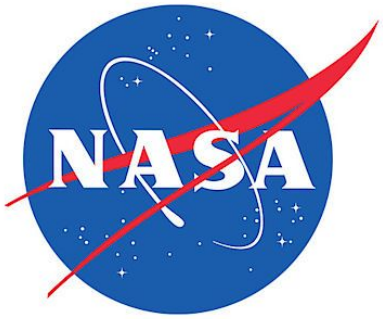
4. Procesamiento de datos
   datos numéricos, gráficos

Características

- ✔ Lenguaje interpretado
- ✔ Puede ser interactivo
- ✔ Fácil de aprender
- ✔ Desarrollo rápido de interfaces
- ✔ Open source
- ✔ Muchas bibliotecas disponibles
- ✔ Sintaxis permite un código legible

… etc

¿Quiénes usan Python?



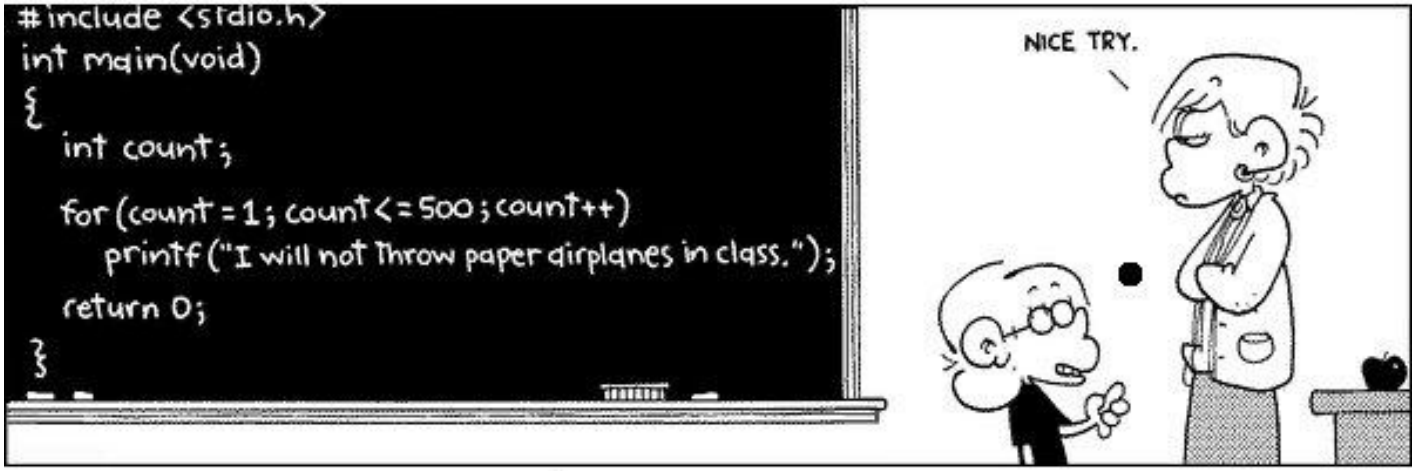
Lenguajes de programación más populares

Rank	Language	Type	Score
1	Python	web, desktop, embedded	100.0
2	Java	web, mobile, desktop	95.4
3	C	mobile, desktop, embedded	94.7
4	C++	mobile, desktop, embedded	92.4
5	JavaScript	web	88.1
6	C#	web, mobile, desktop, embedded	82.4
7	R	desktop	81.7

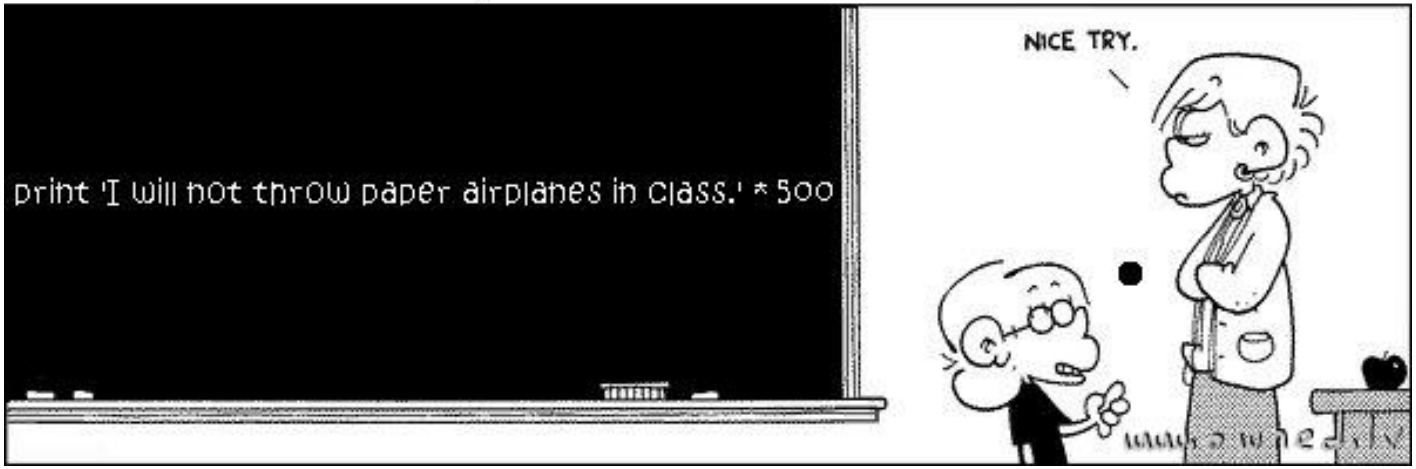
Fuente: IEEE Spectrum, 2021

Let's go!

C version



Python version



Sintaxis

Identificadores

- Letras, underscore (_) y números
- No usar palabras reservadas
- Nombre de clases comienzan con mayúscula (convención)

Líneas e indentación:

- Bloques se delimitan con indentación de líneas
- No se utilizan llaves {} ni punto y coma ;
- Comentarios con #

Operadores

- Numéricos: +,-,*,/,**,%,//
- Asignación: =, +=, -=, *=, /=
- Comparación: ==, !=, <,>,<=,>=
- A nivel de bits: &, |, ^, ~, >>,<<
- Lógicos: and, or, not
- Membrecía: in, not in
- Identidad: is, is not

Tipos de datos

- Números(*): int, float, long, complex, bool
- Secuencias: str (*), list, tuple (*), dict, set (*)
- Clases
- Funciones

¡Todo tipo de datos es un objeto!

(*) Inmutable

Estructuras de control de flujo

Condicionales:

- if, elif, else

Ciclos:

- for, while, break, continue

Funciones de python:

- zip, range, enumerate

Algunas funciones estándares

Built-in Functions				
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

Funciones

- Estructura general:

```
def nombre (parametros):
    '''
     documentacion de la funcion #opcional
    '''
        #cuerpo de la función
        return expresión #opcional
```

- Llamada a función:

```
nombre(parametros)
```

Clases y objetos

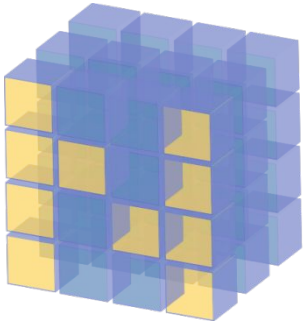
- Estructura general:

```
class Nombre (object):
    ‘’’
     documentacion de la clase #opcional
    ‘’’
    def __ init__ (self, parametros_clase):
        self.atributo = parametro
    def metodo (self, parametros):
        #Cuerpo de la función
                return expresión #opcional
```

- Creación de un objeto:

  objeto = Nombre(parametros_clase)

Procesamiento de datos numéricos y gráficos



NumPy

Pandas

Procesamiento de datos numéricos y gráficos

- Arreglos (ejemplo 4 filas, 4 columnas):

(fila 0, col 0)		...	(fila 0, col 3)
...		...	
		...	
(fila 3, col 0)		...	(fila 3, col 3)

- Tipos de gráfico