

# Introducción a la programación en Python

Emilio Martínez-Núñez  
Departamento de Química Física  
Facultade de Química

# ¿Por qué aprender a programar?

**-Es divertido:** Es como un puzzle en el que debes encajar ciertas piezas de una forma específica para conseguir el resultado deseado.

**-Aumenta la productividad:** Un simple script hace el trabajo en segundos y puedo dedicar ese tiempo ahorrado a cosas más productivas o a vagar en condiciones. Python como para hacer tareas rutinarias de forma automática.

**-Como programador, no te faltará el trabajo:** 900.000 puestos de empleo que el sector digital demandará en la Unión Europea en 2020. En EE.UU. oferta de 570.000/año vs 50.000 graduados en ciencias de la computación/año.

# ¿Por qué aprender a programar?

- Permite crear cosas nuevas:** con conocimientos muy básicos, podrías convertir una idea en un gran proyecto tecnológico. Tal como lo dijo Mark Zuckerberg, *“la programación te permite crear algo totalmente nuevo de la nada”*. ¡Una ventaja que solo pocas profesiones en el mundo te brindan!
- Permite explotar tus habilidades creativas** y ejercitar tu cerebro al máximo, pensando de manera lógica.
- Permite trabajar desde la comodidad de tu hogar:** Si tu empleador es flexible con el horario laboral y lugar de trabajo, esto te permitirá realizar tus funciones desde casa.

# Python

## Introducción

-Código máquina vs programación de alto nivel.

```
010 ! Hello world in Assembler for the HP-85
020 NAM HELLO
030 DEF RUNTIM
040 DEF TOKENS
050 DEF PARSE
060 DEF ERMSG
070 DEF INIT
100 PARSE BYT 0,0
110 RUNTIM BYT 0,0,377,377
120 TOKENS BYT 377
130 ERMSG BYT 377
140 !
150 INIT LDM R26,=MSG
160 ADMD R26,=BINTAB
170 LDM R36,=12D,0
180 JSB =OUTSTR
190 RTN
200 MSG ASC "Hello World!"
210 BINTAB DAD 101233
220 OUTSTR DAD 35052
300 FIN
```

```
# Hello world in Python 3 (aka Python 3000)
print("Hello World")
```

-Lenguajes compilados vs lenguajes interpretados.

# Python

## Introducción

- Diseñado por **Guido van Rossum en 1991** mientras desarrollaba ABC (lenguaje experimental). Hay dos ramas: 2.x y 3.x
- Muy compacto.** Más corto que sus equivalentes en C.
- Muy legible.** Sintaxis elegante y de lectura fácil.
- Entorno interactivo,** que facilita la programación y la realización de pruebas.



# ¿Por qué Python?

## 1. Se imparte en varias asignaturas del Grado en Química (y en otros grados)

**-Estadística e informática para químicos:** 6 sesiones prácticas de 2 horas cada una + 2 horas de introductorias. Tutorial con ejercicios y pequeños programas de uso en otras asignaturas.

**-Matemáticas I y II:** Sage (extensión de Python a las matemáticas).

**-Química Física I:** en parte de las sesiones prácticas se usa Python para tratamiento de datos.

**-Química Física III:** en el futuro para tratamiento de datos. En el presente hay una Wiki:

<https://rxnkin.usc.es/index.php/Thermodynamics>

# ¿Por qué Python?

## 2. Es un proyecto open source

### -Es **open source**:

Los programadores pueden leer, modificar y redistribuir el código fuente de un programa, lo que permite que éste evolucione, se desarrolle y mejore.

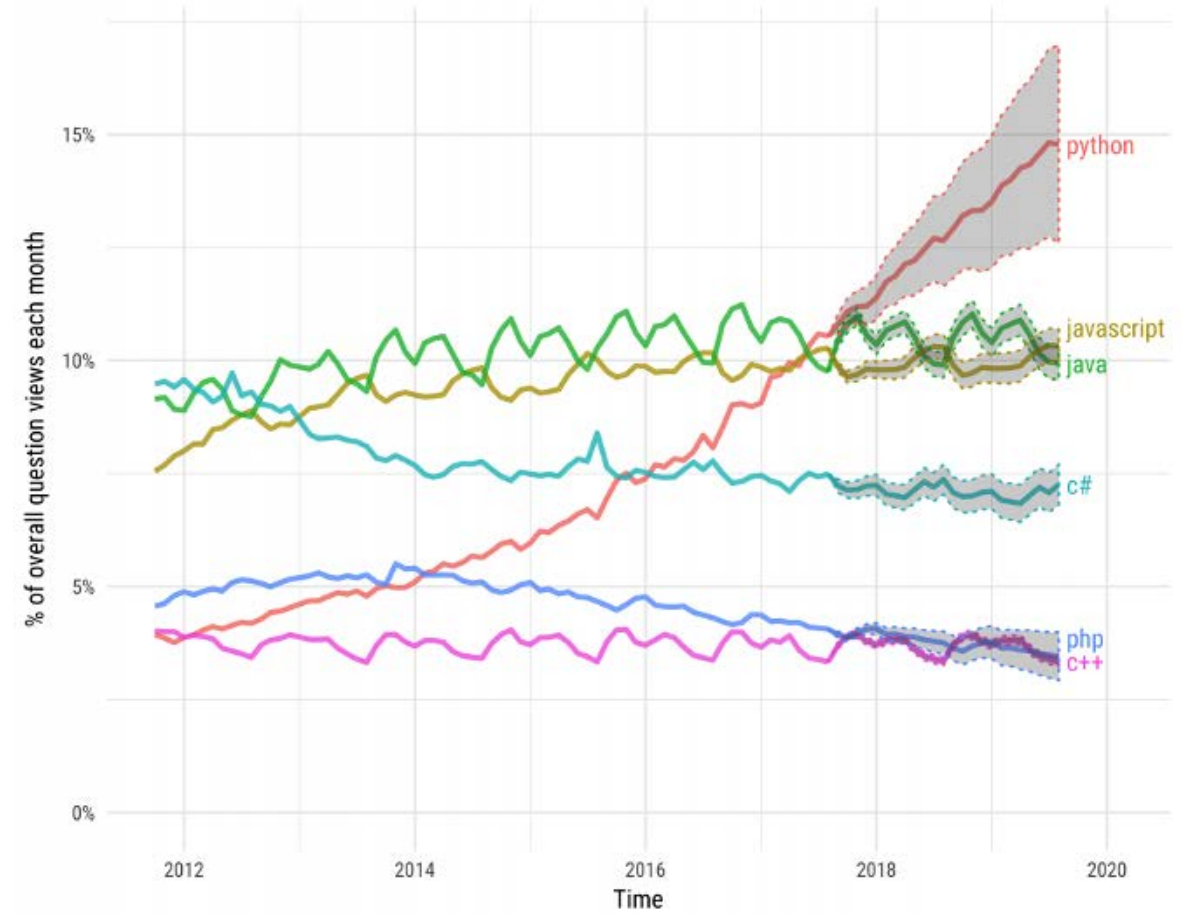
Los usuarios lo adaptan a sus necesidades, corrigen sus errores con un tiempo de espera menor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software.

# ¿Por qué Python?

## 3. Es uno de los lenguajes de programación más populares hoy en día

-No hay una métrica única que mida la popularidad.

-Popularidad basada en el tráfico de datos procedente de EE.UU., RU, Alemania y Canada en Stackoverflow.





# ¿Por qué Python?

## 4. Grandes compañías lo usan

-Uber, PayPal, Google, Facebook, Instagram, Netflix, Dropbox, Youtube, Redditec...

-Intel, HP, NASA, JP Morgan Chase, Goldman Sachs (machine learning)

-Pixar, Industrial Light&Magic

# ¿Por qué Python?

## 5. Machine learning

-Redes neuronales entrenadas sirven para moderar redes sociales a conducir coches.

-El campo ha avanzado mucho en los últimos años gracias a la mejora en el procesamiento de datos.

Ejemplos: [Cancer de pulmón](#), [retrosíntesis](#)

-¿Qué tiene que ver con Python? Un montón. Python es el lenguaje por excelencia para machine learning. La plataforma de [Google TensorFlow](#) trabaja fundamentalmente con Python. Casi cualquier curso sobre redes neuronales usa Python. El análisis de datos requerido por machine learning va muy bien con la estructura de Python y sus librerías. [scikilearn](#)

# ¿Por qué Python?

## 6. Está muy bien documentado

-Debido a su popularidad Python dispone de una gran ayuda online a cualquier nivel. Un gran número de tutorials para principiantes.

-Incluso [Stack Overflow](#) proporciona ayuda a nivel principiante. Por otro lado, programadores experimentados también disponen de ayuda.

# ¿Por qué Python?

## 7. Lo demandan las empresas

-10 lenguajes de programación más demandados por las empresas:

1. Java
2. SQL
3. JavaScript
4. Python
5. HTML
6. CSS
7. C#
8. C++
9. Ruby
10. PHP

# Escribamos un algoritmo

## ¿Qué es un algoritmo?

-Es el conjunto de pasos orientado a la consecución de un objetivo. Ejemplo: receta de cocina.

## Características de un algoritmo:

- Ha de tener 0 o más datos de entrada
- Debe proporcionar 1 o más datos de salida
- Cada paso se debe definir con detalle
- Ha de ser finito, efectivo y eficiente.

# Asignación de variables

No es una ecuación:

`a = expresi on`

`a = 10`

`a = a+1`

`a = i nt(i nput(“Dame un numero: “))`

# Escribamos un algoritmo

Ejemplo:

Media de tres números:

- 1) Solicitar el primer número.
- 2) Solicitar el segundo número.
- 3) Solicitar el tercer número.
- 4) Sumar los tres números y dividir por tres.
- 5) Mostrar el resultado.

[Spyder](#)

```
a = int(input(' Dame el primer numero: '))  
b = int(input(' Dame el segundo numero: '))  
c = int(input(' Dame el tercer numero: '))  
media=(a+b+c)/3  
print(' La media es', media)
```

# Librerías en Python

```
from numpy import sqrt  
rai z = sqrt(4.)  
pri nt (rai z)
```

```
i mport numpy as np  
rai z=np. sqrt(4.)  
pri nt (rai z)
```



# Funciones en Python

```
def suma(a, b):  
    c=a+b  
    return c
```

```
a=1
```

```
b=2
```

```
print(suma(a, b))
```

# Bucles en Python

```
s=0
for i in range(3):
    s=s+i
print(s)
```

```
-----
for x in 'hol a que tal':
    print(x)
```

Ver funciones y for-in en la carpeta correspondiente

# Listas en Python

```
mi_lista=[]  
for i in range(4):  
    mi_lista.append(i)  
print(mi_lista)  
print(len(mi_lista))
```

# Cálculo simbólico en Python

```
from sympy import symbols, diff
x, y=symbols(' x, y' )
eq=x**2+2*y
deri =diff(eq, x, 1)
print("f(x, y) =", eq)
print("f(x, y) +2x=", eq+2*y)
print("df/dx=", deri )
```

Ver lectura de datos y regresión lineal