

# **INTRO A CIENCIA DE DATOS:**

**Conceptos fundamentales de  
Python para Ciencia de Datos**

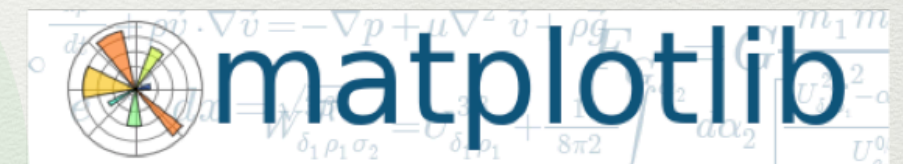


# ¿Por qué aprender Python para Ciencia de Datos?

- Open source, comunidad muy activa y es utilizado por distintos equipos dentro de las empresas
- Librerías poderosas para el análisis de datos
- Curva de aprendizaje moderada, fácil de leer
  - Valor del Tiempo Data Scientist > Tiempo de CPU
- Mejor desempeño que R, Matlab
  - Glue: C, C++, FORTRAN (por ej. Cython)



# Principales Librerías de Python



IP[y]: IPython  
Interactive Computing

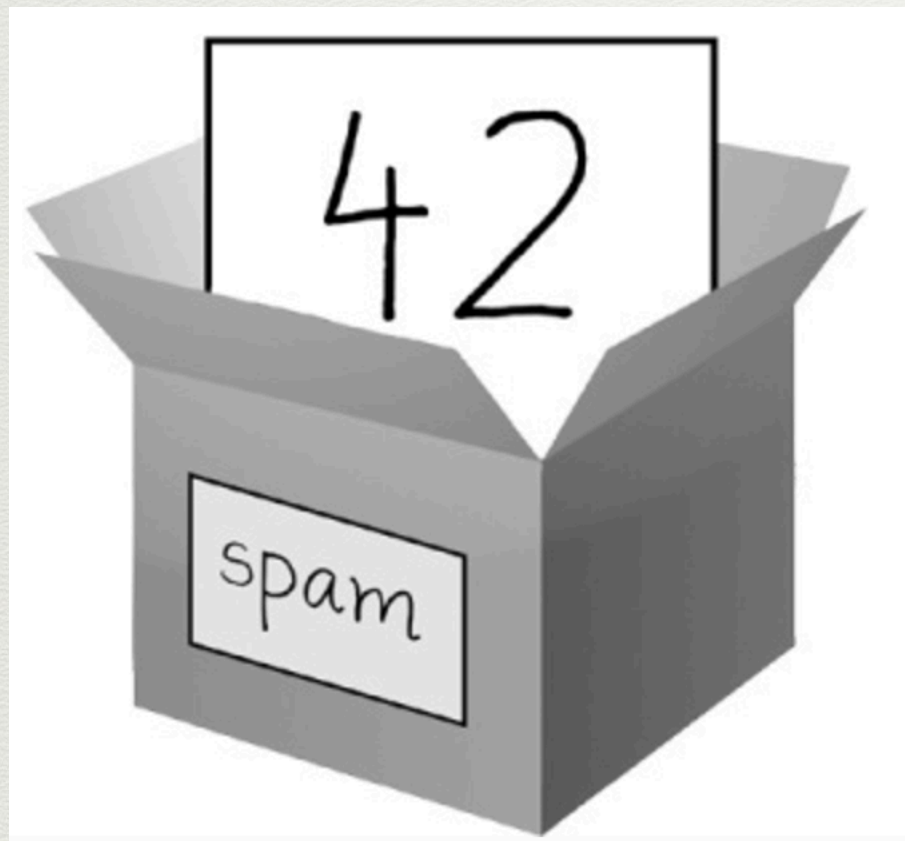




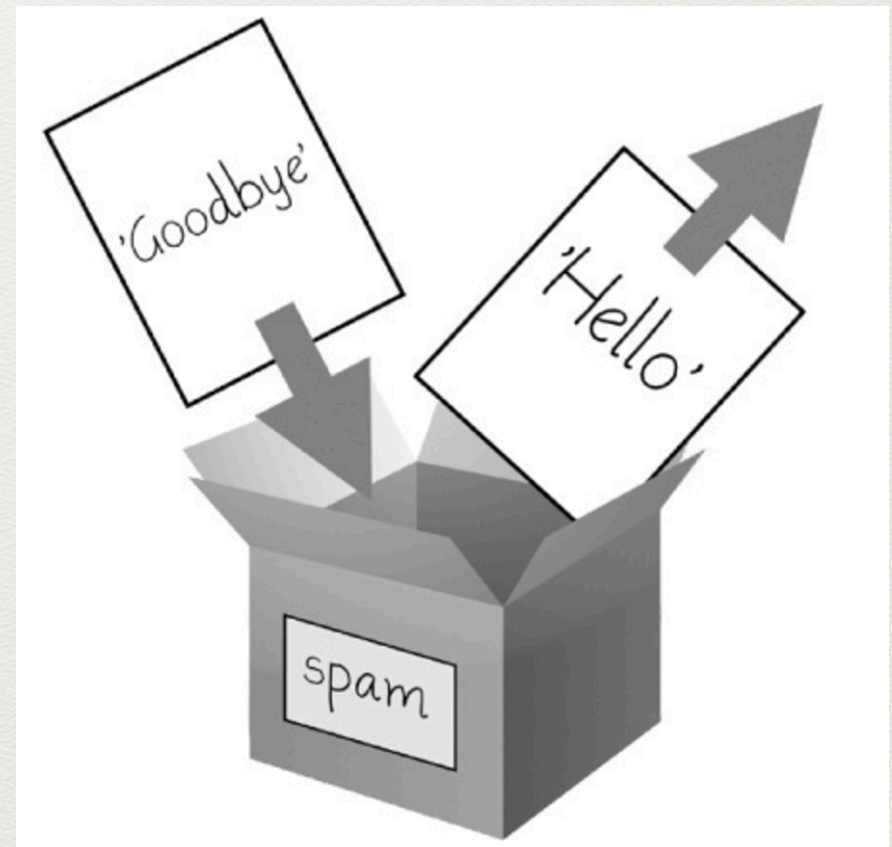
# Variables en Python

- Las variables permiten guardar un valor o conjunto de valores en memoria y asignarles un nombre

Asignar una variable



Re-asignar una variable





# Variables en Python

- Las variables en Python asignan un nombre que hace referencia a un lugar en memoria
- Entonces modificar los datos contenidos en “la caja” (en la memoria) modifica los valores obtenidos para cualquier otro nombre de variable que apunte hacia esa misma caja (lugar en memoria)

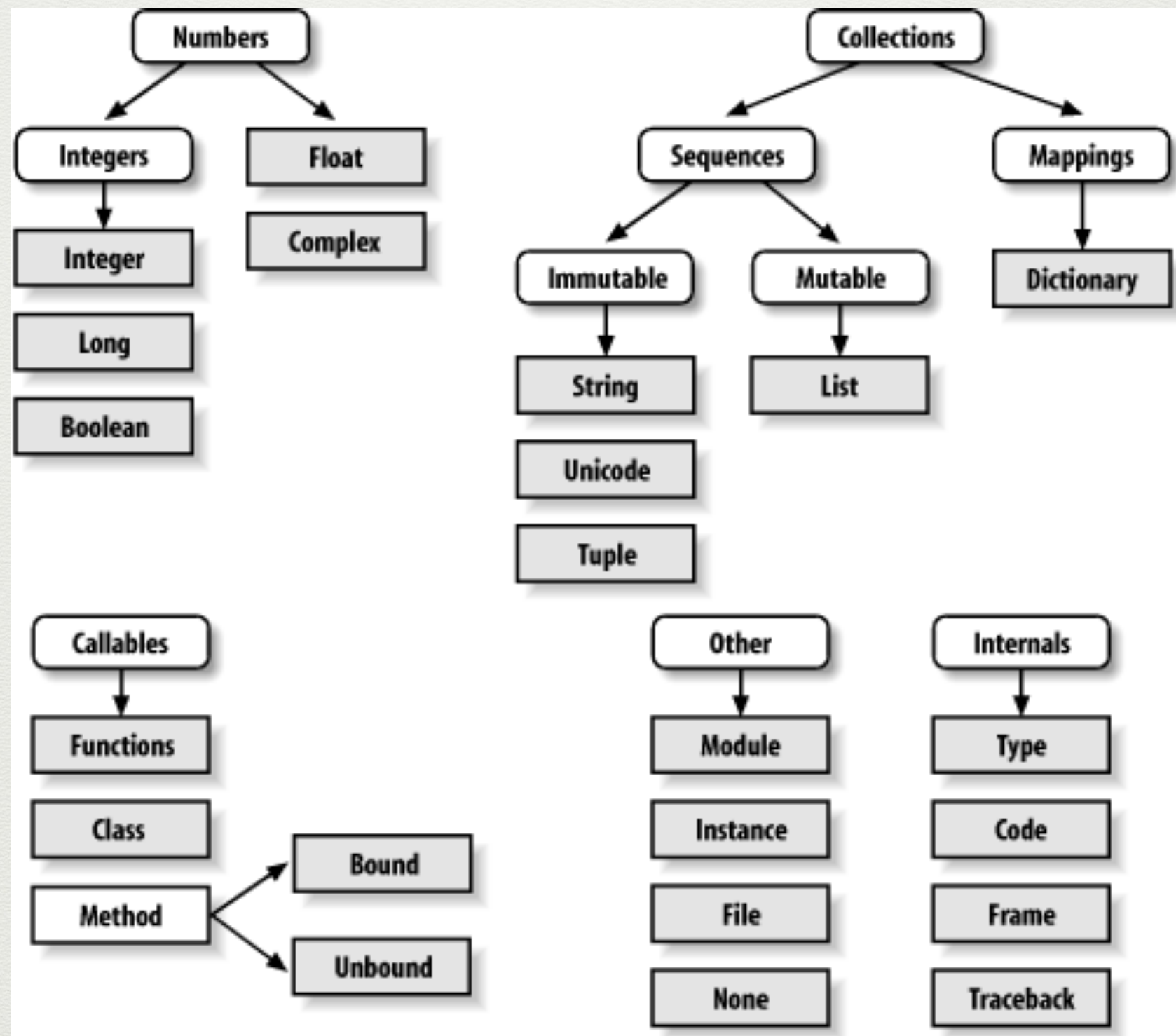


# Todo es un objeto en Python

- Cada objeto en Python tiene un “tipo” y “datos internos” asociados
- De acuerdo al “tipo” se definen:
  - Los *Atributos*: características del objeto
  - Los *Métodos*: lo que el objeto puede hacer



# Tipos de objetos en Python





# Ambientes de Desarrollo

- iPython (2001) mejoró la interfaz gráfica para manejar Python desde la terminal
- Jupyter es una evolución de iPython creada en 2014
  - El nombre proviene de **J**ulia, **P**ython y **R**
  - Está basado en notebooks: documentos interactivos para código, texto y visualizaciones
  - Jupyter interacciona con kernels que permiten utilizar múltiples lenguajes de programación



# Ambientes de Desarrollo

- Los scripts permiten guardar una serie de instrucciones.
- En el caso de Python estas instrucciones se leen línea por línea (i.e., Python es un lenguaje interpretado)
- Los scripts de Python tienen la extensión .py
- Se utilizan editores de texto tales como Atom o Sublime para generar estos Scripts



# Fundamentos de Python

- **Indentación:** Python pondera la indentación sobre el uso de llaves a diferencia de otros lenguajes de programación (Java, C++, JavaScript)
  - Es estándar utilizar 4 espacios para indentar (un tab)
- **Comentarios:** Se pueden escribir líneas de código en Python para explicar mejor tu programa con el signo de #
- **Importar librerías y módulos:** Por cuestiones de eficiencia las librerías no incluidas en la librería estándar de Python se deben importar antes de poder utilizarlas



# Operadores de Comparación y Lógicos

- **Comparación.** Estos operadores comparan los valores de ambos lados y determinan si la relación planteada es verdadera o falsa
  - $>, >=, <, <=$
  - $==, !=$
- **Lógicos.** Evalúan las relaciones entre valores lógicos (o booleanos) e indican si la relación es verdadera o falsa
  - and, or , not



# Control de Flujo

- **Condicionales:** if, elif, else
- **Ciclos:** for, while
  - La función **range**(start, stop, [step]) es útil para usar ciclos pues permite generar progresiones de números
  - Keywords para loops: break, continue, pass
- **¿Qué es un diagrama de flujo?**



# Funciones

- ¿Qué es una función?
  - Diferencia entre una función y un método
- ¿Por qué escribimos funciones?
- Dos momentos para una función: definición y llamar la función
  - ¿Qué son los parámetros y el return? (Input/Output)
- Ingresar parámetros por default y parámetros opcionales



# **INTRO A CIENCIA DE DATOS:**

**Conceptos fundamentales de  
Python para Ciencia de Datos**