

# Funciones Built-in

Antonio Espín Herranz

# Funciones Built-in

- Funciones propias del lenguaje Python.
- Sintaxis: función([parámetros])
- Enlace a documentación:  
<https://docs.python.org/es/3.8/library/functions.html>

# Funciones Built-in

- **filter / reduce / map** → Lo vemos en programación funcional.
- **zip**: Mezcla dos o mas colecciones. El objeto devuelto se puede convertir en lista con la función list.
  - Ideal para construir diccionarios: `d = dict(zip(claves, valores))`
- **all**: Devuelve True si todos los elementos de la colección son True. Un elemento que sea un número será True. Un False, None o cadena vacía serán False.
- **any**: idem de la anterior pero es suficiente con que haya algún elemento a True.
- **round(num, [dígitos])**: Redondear a n decimales: `n = round(5.756, 2) → 5.76`
- **input([prompt])**: Para leer datos de teclado. Ojo, siempre como texto → conversiones de texto a otros tipos. Utilizar las funciones: `int()`, `float()`, `complex()`.
  - Se puede añadir un prompt de forma opcional.

# Funciones Built-in

- **open**: Para abrir un fichero. Devuelve el fichero abierto o lanza una excepción si no existe, o pudiera estar bloqueado por el S.O.
- **pow**(base, exp, [mod]): Devuelve la base elevada al exponente. Se puede calcular también el módulo utilizando el tercer parámetro, en este caso en vez de devolver la potencia devuelve el módulo de la potencia calculada.
  - $\text{pow}(2,10,4) \rightarrow 2^{**}10 = 1024 \bmod 4 = 0$
- **super**([type [, object-or-type]]): Para hacer llamadas al constructor o métodos heredados de una clase padre.
- **min / max**: Devuelve el mínimo / máximo de una colección.

# Funciones Built-in

- **sorted(iterable, key=None, reverse=False)**: Ordena una colección y la devuelve como una lista ordenada.
- **int / float / complex**: Para convertir de texto a: entero, real o complejo. La función int espera un texto que represente un número entero en base 10, se puede indicar en otras bases: `int('12F', 16)`
- **sum(iterable)**: Sumar los elementos de un iterable.
- **len(iterable)**: Número de elementos de un iterable.