

🛑 🧐 ¿Qué es un tipo?

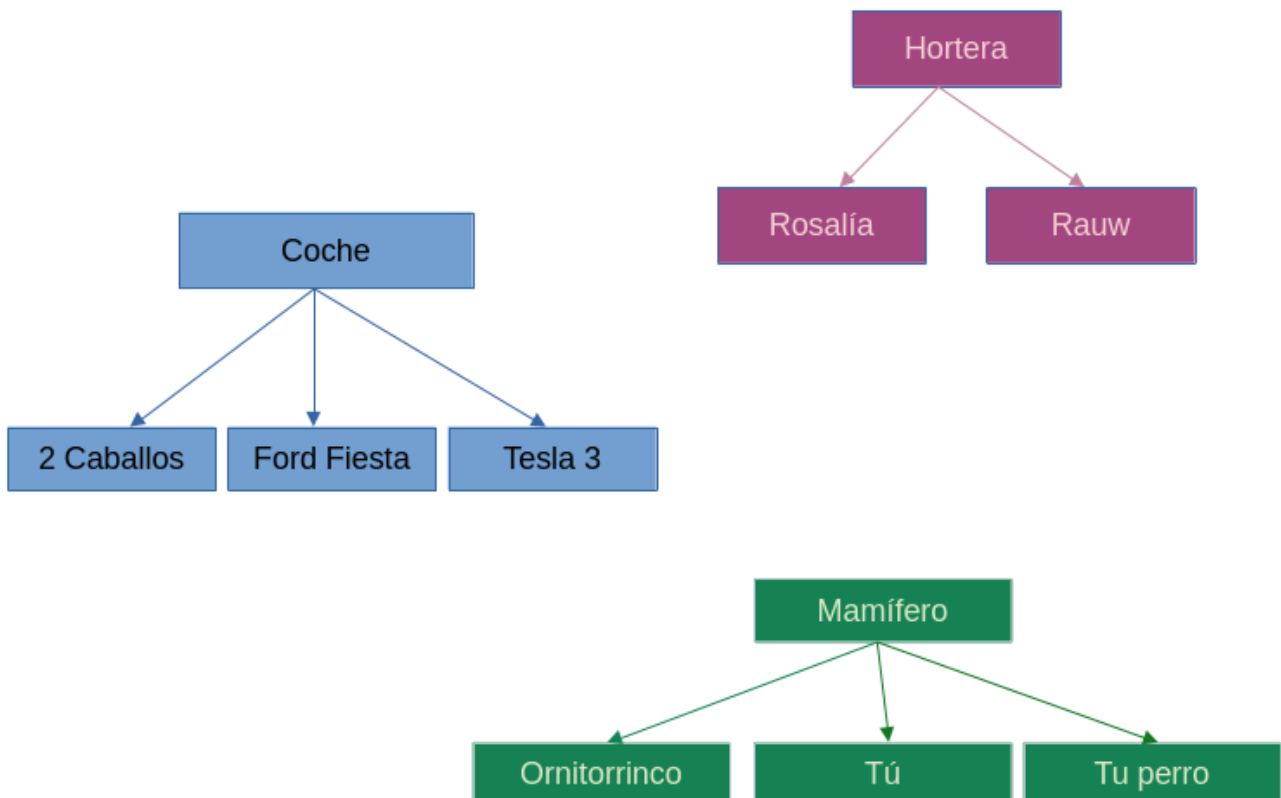
Un tipo es un concepto que se usa para agrupar y organizar cosas que tienen algo (o mucho) en común. Por ejemplo:

- Un ornitorrinco, un perro y tú, sois *mamíferos*. *Mamífero* es el **tipo** de animal que sois (con todo respeto).
- Un Ford Fiesta, un Tesla modelo X y un Dos Caballos son *coches*. *Coche* es el **tipo** de máquina que son.
- Rosalía y Rauw son *horteras*. *Hortera* es el **tipo** de Persona que son.

Un **tipo** es un conjunto de cosas

A esas cosas las llamaremos **instancias del tipo**

1. Tú, el ornitorrinco y el perro, sois **instancias** del **tipo** Mamífero.
2. El Ford Fiesta, el Tesla y el Dos Caballos son **instancias** del **tipo** Coche.
3. Rosalía y Rauw (que Dios los perdone) son **instancias** del **tipo** Hortera.



Los tipos, pueden a su vez ser instancias de otros tipos

No es nada raro y lo vemos todos los días.

1. Mamífero es una *instancia* del **tipo** Animal.

2. Coche es una *instancia* del **tipo** Máquina.
3. Hortera es una *instancia* del **tipo** Persona, que a su vez es una **instancia** del **tipo** Animal.

Ni momento Inception ni leches, es bien normal.

Tipos en Programación

Los Tipos se usan en programación para organizar los datos, según su, ehem, tipo.

- "hola", "Rigoberto", "" y "3" son todos del tipo Cadena.
- 3, -10, -15937463 son todos del tipo Número

Número de Instancias

Los tipos que vimos al principio (Animal, Coche y Hortera) tienen todos un *número finito de instancias*. Es decir, no hay infinidad de modelos de coches, ni de animales, ni tampoco (aunque parezca mentira) de horterías.

También hay tipos que contienen una *infinidad de instancias*. Por ejemplo, el tipo *Cosas que no son verdes*: podríamos estar hasta el fin de los tiempos recitando cosas que no son verdes (no lo haremos).

En programación esto también es común:

1. ¿Cuántas instancias tiene el tipo *Cadena*? Olvídate de limitaciones de memoria.
2. ¿Cuántas instancias tiene el tipo *Número*? Olvídate de limitaciones de memoria.

Dentro de poco veremos un tipo que sólo tiene 2 instancias.

Tipos de Python vistos hasta ahora

Hagamos un repaso.

Cadenas

En realidad, en Python se llaman `str` (de String).

Un tipo es un conjunto de valores y las operaciones que se pueden hacer sobre ellos

Las operaciones que se pueden hacer sobre las cadenas son los operadores que ya hemos visto y un más que veremos ahora.

- `+` : concatenar
- `*` : extender
- `[]` : extraer elementos
- `[:]` : *slicing* o fileteado.

Los dos últimos los veremos en un lab.

En realidad, son varios tipos, por razones debidas al hardware de los procesadores y que no vienen muy a cuento.

[illegible]

Las operaciones estos tipos son las que ya conocemos:

- + , -
_ * /

Pues tenemos una función del ámbito global llamada `type()` que hace precisamente eso.

- "hola"
- ""
- 0.0
- math.pi
- math.e

In [2]:

```
def succ(n):  
    return n + 1
```

In [3]:

```
type(succ)
```

Out[3]:

function

😲 **function** es un tipo y ¡todas las funciones (las tuyas o las de Python) son instancias de dicho tipo!

Las funciones no tienen nada de mágico, son un tipo más de Python.

Somos la Casa Lambda. No hay llamada que no atendamos.

Operaciones sobre las funciones

Habíamos dicho que,

Un tipo es un conjunto de valores y las operaciones que se pueden hacer sobre ellos

Pues bien, ¿cuales son las operaciones que se pueden hacer sobre las funciones?

😬 **La llamada** Toda función es *llamable* (*callable* en Inglés), y entiende el operador `()` para efectuar esa llamada.



¿Qué hemos aprendido?

- Los tipos sirven para agrupar valores
- Un tipo está definido por sus instancias y las operaciones que se pueden hacer sobre ellas
- Hay tipos con un número infinito de instancias y otros con un número finito
- Los números en Python son los tipos `int`, `float` y uno raro para números complejos.
- Las cadenas en Python son el tipo `str`
- ¡Las funciones son instancias del tipo `function` !
- Tienes que leer *Dune*.

En el siguiente capítulo veremos un nuevo tipo que sólo tiene dos instancias y es vital: los booleanos.