

Más métodos especiales

Hemos visto que las clases admiten dos métodos especiales: `__init__` y `__str__`. No son los únicos métodos especiales. Podemos hacer que las clases se comporten de modo similar a los tipos de datos nativos de Python definiendo muchos otros métodos especiales. He aquí unos pocos:

- `__len__(self)`: Permite aplicar la función predefinida `len` sobre objetos de la clase. Debe devolver la «longitud» o «talla» del objeto. En el caso de colas y conjuntos, por ejemplo, correspondería al número de elementos. Si *A* es un *Conjunto*, podríamos usar `len(A)` si antes hubiésemos definido el método `__len__`.
- `__add__(self, otro)`: Permite aplicar el operador de suma (+) a objetos de la clase sobre la que se ha definido. Si, por ejemplo, *A* y *B* son conjuntos, la expresión `C = A + B` permite asignar al nuevo conjunto *C* la unión de ambos.
- `__mul__(self, otro)`: Permite aplicar el operador de multiplicación (*) a objetos de la clase sobre la que se ha definido.
- `__cmp__(self, otro)`: Permite aplicar los operadores de comparación (<, >, <=, >=, ==, !=) a objetos de una clase. Debe devolver -1 si *self* es menor que *otro*, 0 si son iguales y 1 si *self* es mayor que *otro*.

Podemos, por ejemplo, definir `__cmp__` en *Persona* para que devuelva -1 cuando la edad `self.edad` es menor que `otro.edad`, 0 si son iguales y 1 si `self.edad` es mayor que `otro.edad`. Si *juan* y *pedro* son personas, podremos compararlas con expresiones como `juan < pedro` o `juan != pedro`.

Consulta la documentación de Python si quieres conocer todos los métodos especiales que puedes definir en tus clases. Tus programas pueden ganar mucho en elegancia si defines los métodos apropiados para cada clase.

7.4.1. Videoclub básico

El videoclub tiene un listado de socios. Cada socio tiene una serie de datos:

- dni,
- nombre,
- teléfono,
- domicilio.

Por otra parte, disponemos de una serie de películas. De cada película nos interesa:

- título,
- género (acción, comedia, musical, etc.).

Supondremos que en nuestro videoclub básico solo hay un ejemplar de cada película.

Empecemos definiendo los tipos básicos con sus métodos especiales: el constructor `__init__` y el conversor a cadena `__str__`:

```
videoclub.py
1 class Socio:
2     def __init__(self, dni, nombre, teléfono, domicilio):
3         self.dni = dni
4         self.nombre = nombre
5         self.teléfono = teléfono
6         self.domicilio = domicilio
7
8     def __str__(self):
9         return 'DNI:_{0}\nNombre:_{1}\nTeléfono:_{2}\nDomicilio:_{3}\n' \
10             .format(self.dni, self.nombre, self.teléfono, self.domicilio)
11
```