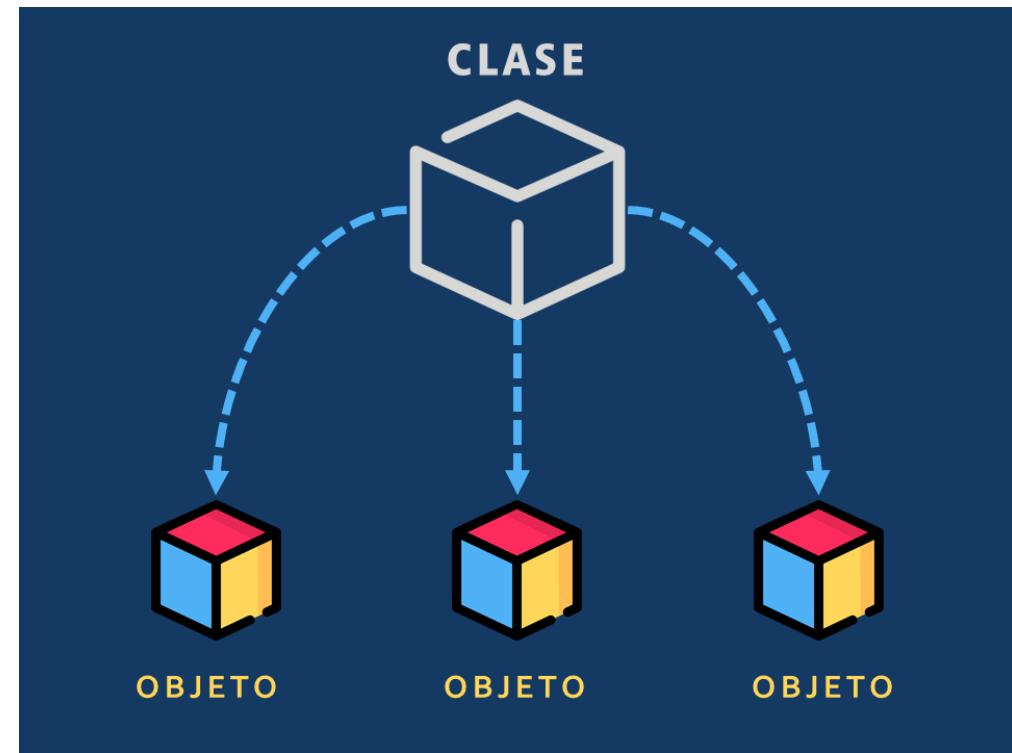


Clases y objetos en Python



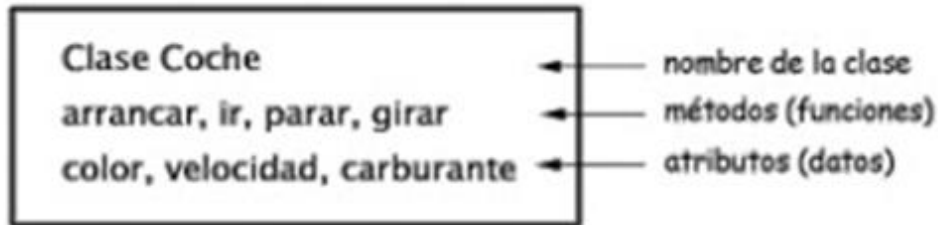
Programación orientada a objetos

Es un paradigma (estilo) de programación que basa su funcionamiento en objetos, donde cada uno de ellos es una entidad que tiene propiedades, atributos y métodos (funciones) que los transforman.

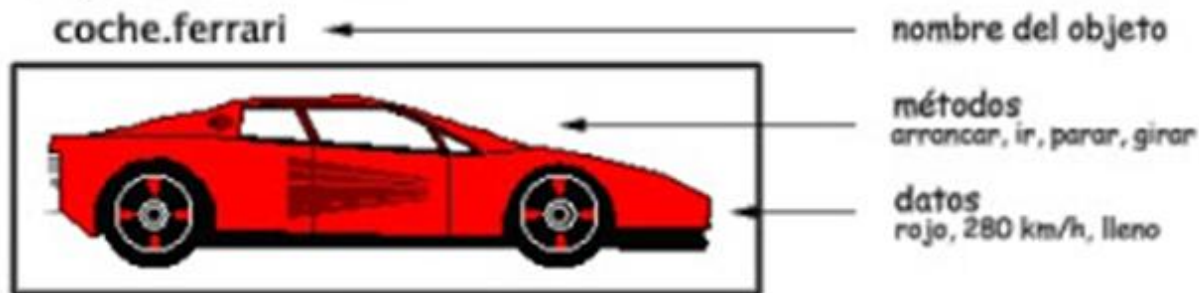


Clases y objetos

Clase: *Coche*



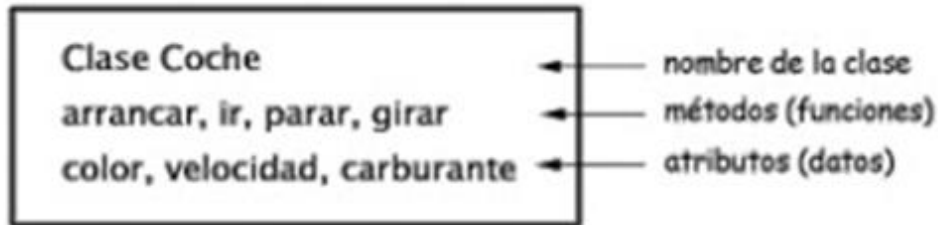
♦ Objeto: *Ferrari*



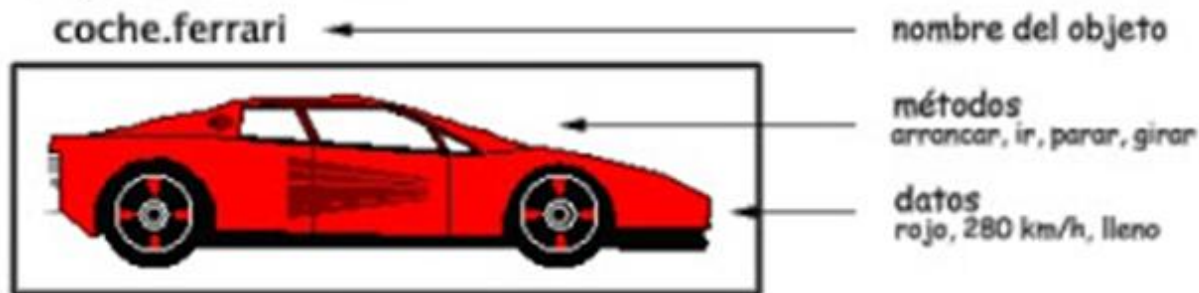
- Una *clase* representa al conjunto de objetos que comparten una estructura y un comportamiento comunes. Una clase es una combinación específica de atributos y métodos y puede considerarse un tipo de dato

Clases y objetos

Clase: *Coche*

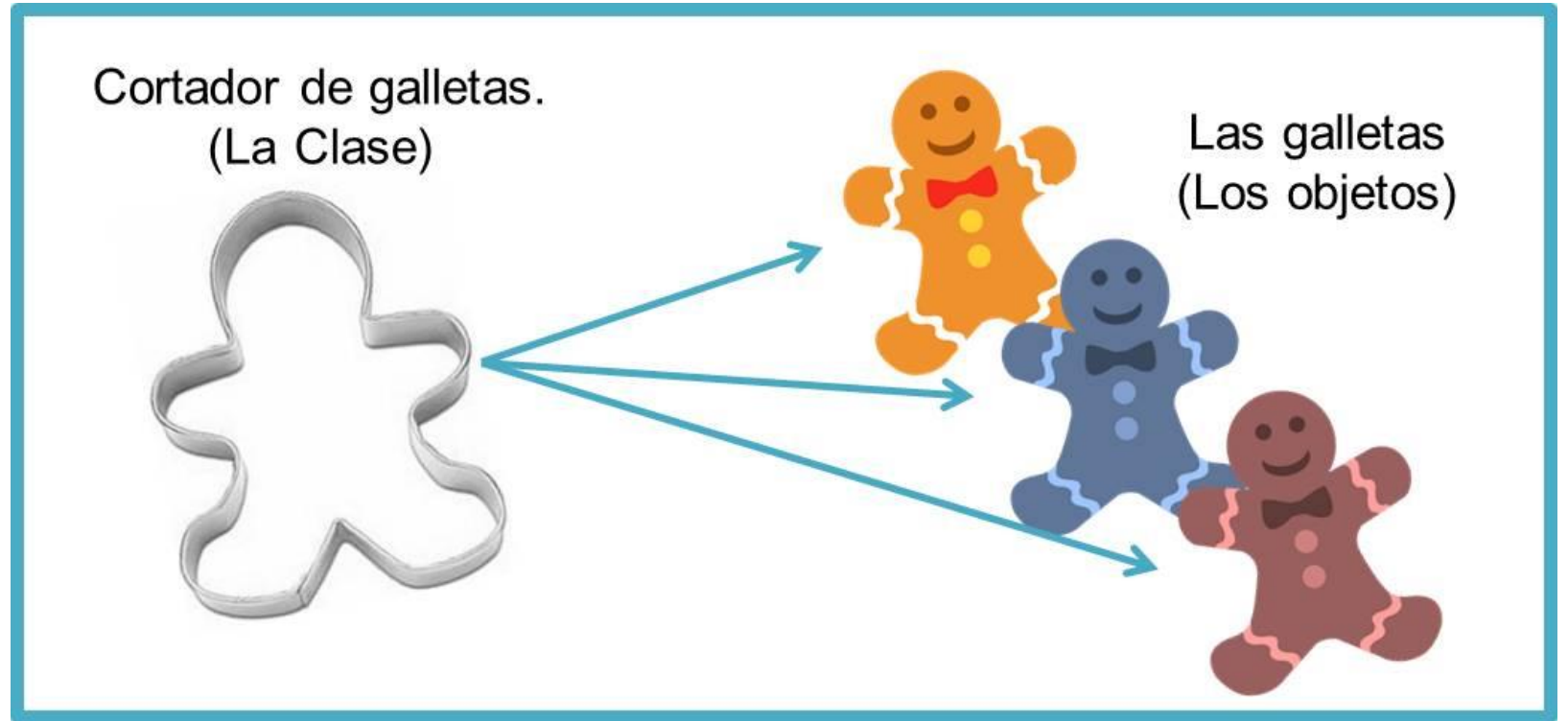


♦ Objeto: *Ferrari*



- Una clase es un tipo de dato definido por el usuario, al crear instancias de una clase, se crean objetos de ese tipo.
- La definición de una clase comienza con la palabra reservada **class** seguido del nombre de la clase. Tener en cuenta que el nombre de la clase sigue las mismas reglas que los nombres de variables en Python.

Clase y objetos



Una clase es una especie de *plantilla o prototipo* de objetos: define los atributos que componen ese tipo de objetos y los métodos que pueden emplearse para trabajar con esos objetos.

Atributos

- Los atributos son las variables o propiedades que definen a una clase
- Son comunes para todos los objetos creados
- Se pueden consultar o modificar con una simple instrucción

```
class Persona():  
    nombre = valor1  
    edad   = valor2  
    ...  
    peso   = valorN
```

Atributos

- Creando un objeto de la clase Persona, y consultando el valor de sus atributos
- También podemos actualizar el valor de los mismo

```
persona1 = Persona()
```

```
persona1.nombre  
>> valor1
```

```
persona1.edad  
>> valor2
```

```
persona1.nombre = nuevo_nombre  
persona2.edad   = nueva_edad
```

Métodos

- Los métodos son las funciones creadas dentro de una clase, se definen con la palabra reservada **def** y cuentan con el mismo formato que las funciones tradicionales. Habitualmente cuentan con el parámetro **self** para referirse a los atributos de la clase.

```
class Persona():  
    nombre = 'Alfredo'  
    edad = 42  
  
    def presentate(self):  
        print('Soy', self.nombre)  
  
p1 = Persona()  
  
p1.presentate()  
>> 'Soy Alfredo'
```


Inicialización

- El inicializador es un método especial, con nombre `__init__`. Es un método particular, se trata como una palabra reservada y lleva doble guión bajo como otras funciones de Python.

```
class Persona():  
  
    def __init__(self, nombre, edad):  
        self.nombre = nombre  
        self.edad = edad  
  
    def presentate(self):  
        print('Soy', self.nombre)  
  
p1 = Persona( 'July', 27 )  
  
p1.edad + 1  
>> 28
```

Implementación

- Nombre de clase es el type:

```
class Punto():
```

- Definir métodos con `def`:

```
def modulo():
```

```
<instrucciones>
```

- Usar `self`, para referirse al scope interno:

```
np.sqrt((self.x^2-self.y^2))
```

- Los métodos son comunes a todas las instancias.

Instanciación

- Instancia es un objeto específico de esa clase:

```
punto1=Punto(1,3)
```

```
punto2=Punto(2,1)
```

- Usar el objeto para acceder a los valores:

```
print(punto1.x)
```

```
punto2.x = 15
```

- Usar el objeto para acceder a métodos:

```
punto1.modulo()
```

Clase



Atributos



Inicialización



Métodos



1

```
class Punto():
```

2

```
    memoria = True
```

3

```
    def __init__(self, x, y):  
        self.x = x  
        self.y = y
```

4

```
    def suma(self, p):  
        z = Punto()  
        z.x = self.x + p.x  
        z.y = self.y + p.y  
        return z
```

Atributos

- Propiedad o característica de un objeto

Métodos

- Son funciones que corresponden a una clase determinada

Clase Taza

- Atributos
 - Color
 - Forma
 - Tamaño

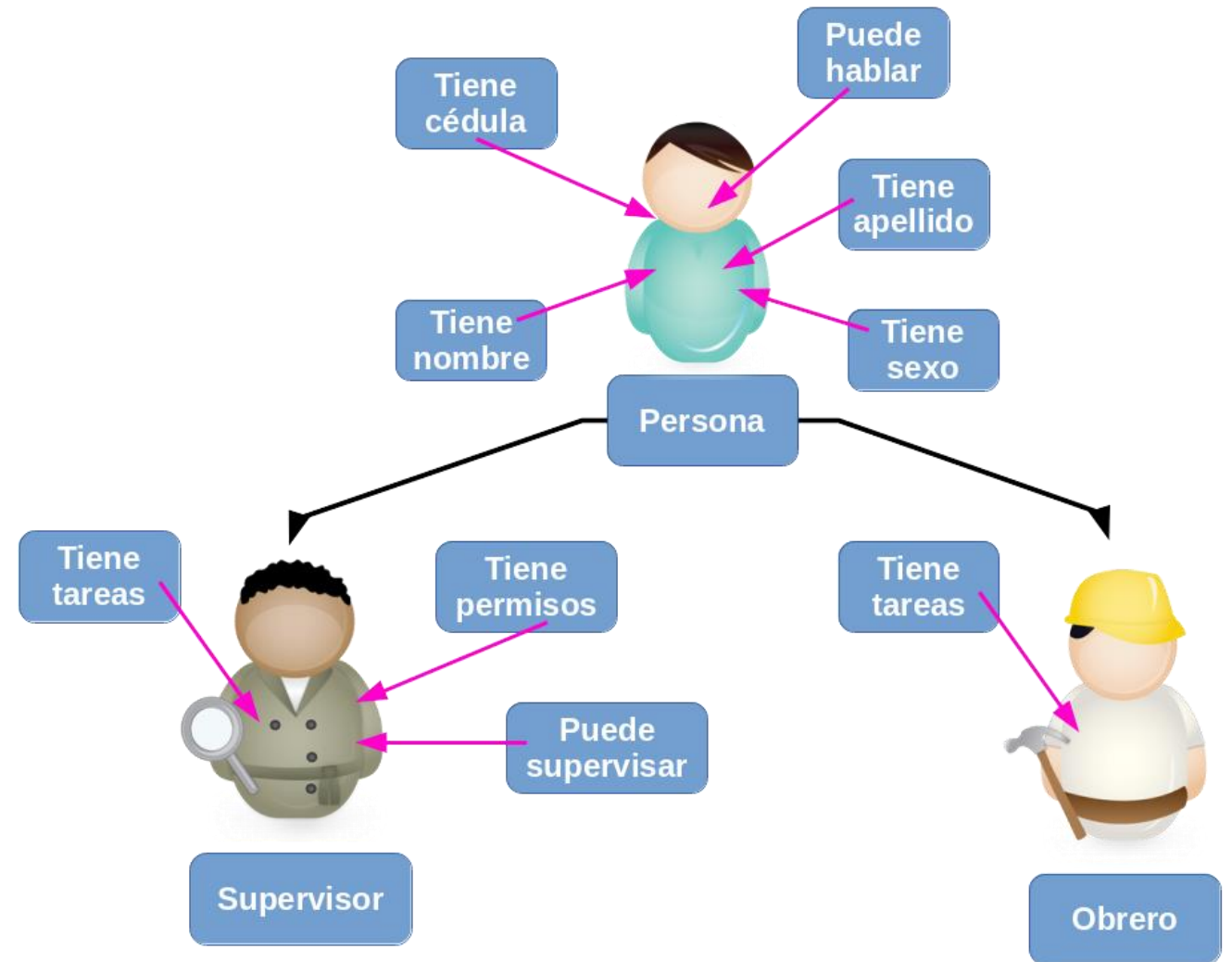


- Métodos
 - Sostenerse
 - Almacenar



Herencia

Implica compartir de una clase sus atributos y comportamientos con otras clases derivadas



Ejercicios

- <https://www.youtube.com/watch?v=c07ZdfoIcnU>
- <https://www.youtube.com/watch?v=0FpdNQ1D5ko>
- <https://www.youtube.com/watch?v=s17K6v5EUYI>
- <https://www.youtube.com/watch?v=X8iXwT9XFvM>