

A close-up photograph of several wooden spindles arranged in a row, likely part of a lathe or a decorative element. The spindles are light brown and have a smooth, rounded, conical shape. They are mounted on a dark metal rod. The background is dark and out of focus.

ESCUELA DE INGENIERÍA

Atributos



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE



Cristian Ruz Ruz

Profesor Asistente Adjunto, Departamento de
Ciencia de la Computación
Facultad de Ingeniería
Pontificia Universidad Católica de Chile

Índice



Atributos



Inicialización de atributos


Introducción

- ¿Qué tipos de atributos podemos usar?
- ¿Cómo inicializamos atributos en Python?

Atributos

Atributos

- **Atributos:** campos o características del **objeto**.
- **Todos los objetos** pertenecientes a **la misma clase** poseen los mismos **nombres de atributos**.
- **Cada instancia** de la misma clase, le **asigna valores** a esos atributos.



Auto
Marca
Modelo
Año
Color
Kilometraje
Ubicación
dueño
conducir(kms)
vender(persona)
pintar(color)

Objeto de la clase Auto

"kia"
"sportage"
2000
"blanco"
145230
(-35.32, -72.41)
"Enrique"

Objeto de la clase Auto

"suzuki"
"grand nomade"
2015
"naranja"
35695
(-33.45, -70.63)
"Carla"

Inicializando atributos en Python

Inicializando atributos: método `__init__()`

```
class Auto:
    def __init__(self, ma, mo, a, c, k):
        self.marca = ma
        self.modelo = mo
        self.año = a
        self.color = c
        self.kilometraje = k
        self.ubicacion = (-33.45, -70.63)
        self.dueño = None

    def conducir(self, kms):
        self.kilometraje += kms

    def vender(self, nuevo_dueño):
        self.dueño = nuevo_dueño

    def leer_odometro(self):
        return self.kilometraje
```

Definición de atributos
Método *inicializador*

Invocación de
método *inicializador*

Creación de instancias de
Auto (objetos de clase Auto)

```
a = Auto("kia", "sportage", 2000, "blanco", 145230)
b = Auto("suzuki", "grand nomade", 2015, "naranja", 35695)
print(a.modelo)
a.color = "azul"
```

Lectura de
atributo de a

Modificación de
atributo de a

Atributos
"privados" {

```
class Auto:
    def __init__(self, ma, mo, a, c,
k):
        self.marca = ma
        self.modelo = mo
        self.año = a
        self.color = c
        self.__kilometraje = k
        self.__ubicacion = (-33.45,
-70.63)
        self.dueño = None

    def conducir(self, kms):
        self.__kilometraje += kms
        self.__modificar_ubicacion()

    def vender(self, nuevo_dueño):
        self.dueño = nuevo_dueño

    def leer_odometro(self):
        return self.__kilometraje

    def __modificar_ubicacion(self):
        print("Calcula nueva
ubicación")
        self.__ubicacion =
```

Encapsulamiento

- Objetos solo deben ser manipulados mediante su **conjunto de métodos**: su **interfaz**.
- Al usar un objeto, no debe importar **cómo** está implementado un método, sino **lo que hace** el método.
- Atributos y métodos que empiezan con _ no **deben** ser llamados desde fuera de la clase (convención).
- Si empiezan con __ **no pueden** ser llamados desde fuera de la clase.

```
a = Auto("kia", "sportage", 2000, "blanco",
145230)
b = Auto("suzuki", "grand nomade", 2015,
"naranja", 35695)
print(a.modelo)
a.color = "azul"
print(a.__kilometraje)
a.__kilometraje += 42
```

Esto provoca un error, pues `__kilometraje` es un atributo privado

Esto está bien pues la modificación se hace dentro de la clase

Síntesis

- Atributos.
- Definición de atributos en Python.
- Lectura y modificación de atributos, encapsulamiento de atributos.

Referencias bibliográficas

- Documentación Python 3.8. Classes. <https://docs.python.org/3/tutorial/classes.html>



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE