

Creado por:

Isabel Maniega

Ejemplo 1:

Conceptos:

- clase: cliente
- init: método init, cimilar a un constructor

(no es un contructor, pero cumple esa función)

```
In [1]: class Cliente:
        def __init__(self):
            self.edad = 42
```

```
In [2]: cliente1 = Cliente()
cliente2 = Cliente()
```

```
In [3]: cliente1.edad
```

```
Out[3]: 42
```

```
In [4]: cliente2.edad
```

```
Out[4]: 42
```

Conclusión:

- No es práctico
- Todos los clientes tendrán esa edad (a no ser que se la cambiemos)

Ejemplo 2

Conceptos:

- cliente --> clase
- init --> es el método init
- nombre, edad, estado_civil y profesión son las propiedades
- Instanciamos...
- los posibles clientes (cliente1, cliente2, cliente3...) --> son los "objetos"
- y el init los iniciarlos

- self indica que:

(el alcance de esas variables (atributos) sea de clase y no de método solamente)

```
In [5]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "María", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")
```

```
In [6]: cliente1.nombre
```

```
Out[6]: 'Juan'
```

```
In [7]: cliente2.nombre
```

```
Out[7]: 'Maria'
```

Ejemplo 3

Concepto:

- Puedo crear más funciones
- Con este ejemplo se comprenderá el objetivo de POO

```
In [17]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

        def metodo_1():
            print("aquí creo el contenido del método 1")

        def metodo_2():
            print("aquí creo el contenido del método 2")

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "María", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")
```

```
In [18]: Cliente.metodo_1()
```

aquí creo el contenido del método 1

```
In [19]: Cliente.metodo_2()
```

aquí creo el contenido del método 2

Ejemplo 4

```
In [20]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

        def metodo_1():
            print("aquí creo el contenido del método 1")

        def metodo_2(x):
            print("aquí creo el contenido del método 2")
            print("el valor de x es: ", x)

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "Maria", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")
```

```
In [21]: Cliente.metodo_2(5)
```

aquí creo el contenido del método 2
el valor de x es: 5

Ejemplo 5

```
In [24]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

        def metodo_1():
            print("aquí creo el contenido del método 1")

        def metodo_2():
            print("aquí creo el contenido del método 2")
            print("el nombre del cliente1 es: ", cliente1.nombre)

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "Maria", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")
```

```
In [25]: Cliente.metodo_2()
```

aquí creo el contenido del método 2
el nombre del cliente1 es: Juan

Ejemplo 6

```
In [26]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

        def tu_metodo(self):
            print("la edad del cliente seleccionado es: ", self.edad)

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "Maria", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")

# escoger el cliente y "tu_metodo"
cliente1.tu_metodo()
```

la edad del cliente seleccionado es: 30

Ejemplo 7

Conceptos:

- Otra forma de imprimirlo
- Recordad que hay varias formas de impresión

```
In [28]: class Cliente:

        def __init__(self, Id, nombre, edad, estado_civil, profesion):
            self.Id = Id
            self.nombre = nombre
            self.edad = edad
            self.estado_civil = estado_civil
            self.profesion = profesion

        def infoCliente(self):
            # Con esta otra forma de impresión
            # print(f"....{self.variable}")
            print(f"El cliente es: {self.nombre}")
            print(f"La edad del cliente seleccionado es: {self.edad}")
            print("El estado civil del cliente es %s" % (self.estado_civil))
            print(f"La profesion del cliente es {self.profesion}")

cliente1 = Cliente(1, "Juan", 30, "soltero", "ingeniero")
cliente2 = Cliente(2, "Maria", 45, "casada", "arquitecta")
cliente3 = Cliente(3, "Carlos", 25, "soltero", "abogado")
```

```
In [29]: cliente2.infoCliente()
```

El cliente es: María
La edad del cliente seleccionado es: 45
El estado civil del cliente es casada
La profesion del cliente es arquitecta

```
In [30]: cliente3.infoCliente()
```

El cliente es: Carlos
La edad del cliente seleccionado es: 25
El estado civil del cliente es soltero
La profesion del cliente es abogado

```
In [31]: cliente1.infoCliente()
```

El cliente es: Juan
La edad del cliente seleccionado es: 30
El estado civil del cliente es soltero
La profesion del cliente es ingeniero

Creado por:

Isabel Maniega