PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

```
class NombreClase:
    pass

objeto_1 = NombreClase()

print (type(objeto_1))
```

```
<class '__main__.NombreClase'>
```

Una clase se crea con la palabra reservada ***class*** seguida del identificador de clase ***CamelCase***

pass permite definir una clase (o función) vacía

Un objeto es la instancia de una clase. Se crea utilizando el nombre de la clase como si fuera una función.

Un ***método de clase*** es una función declarada dentro de la clase.

El ***constructor*** de nuevos objetos se implementa como un método de nombre __init__
Incluye todas las sentencias que se ejecutarán al crear un objeto y permite agregar propiedades al objeto.

```
class NombreClase:
    def __init__(self):
        print("Objeto creado")
        self.valor = 0
        self.__modelo = "A"

    def modifica(self, val):
        self.valor = val
```

El componente de una clase que comienza por __ es privado a la clase

El primer ***parámetro OBLIGATORIO*** de los métodos de una clase representa al objeto instanciado. Generalmente se denomina ***self***. Permite que el objeto pueda acceder a sus componentes.

PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

Se llaman ***Variables de instancia*** a las ***propiedades de los objetos***. Se pueden crear:

- en la inicialización del objeto por el constructor
- en cualquier momento de la vida del objeto.

Diferentes objetos de una misma clase pueden tener diferentes propiedades. Es necesaria una forma de verificar las propiedades que tiene un objeto.

Los objetos de Python, tienen una propiedad predefinida llamada ***__dict__*** (es un diccionario). La variable contiene los nombres y valores de todas las propiedades (variables) que el objeto contiene actualmente.
La función ***hasattr(objeto,'valor')*** nos permite comprobar si ***objeto*** tiene un atributo llamado ***valor***

```
class ExampleClass:
    def __init__(self, val = 1):
        self.primero = val

    def set_second(self, val):
        self.segundo = val

object_1 = ExampleClass()
object_1.tercero = 'A'
```

```
class NombreClase:
    def __init__(self):
        self.__valor = 0
        self.modelo = "berlina"

objeto_1 = NombreClase()
print(objeto_1.__dict__)
```

```
{'_NombreClase__valor': 0, 'modelo': 'berlina'}
```

```
if hasattr(objeto_1, 'modelo'):
    print(objeto_1.modelo)

if hasattr(objeto_1, '_NombreClase__valor'):
    print("Variable de instancia privada")
    print(objeto_1._NombreClase__valor)
```

```
berlina
Variable de instancia privada
0
```

PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

Se llama ***variable de clase*** a la ***propiedad que existe en una sola copia y se almacena fuera de cualquier objeto .***

Al contrario que las variables de instancia, una variable de clase existe aunque no haya objetos de la clase.

En el constructor, se accede a la variable de clase anteponiendo el nombre de la clase.

```
class ExampleClass:
    contador = 0
    def __init__(self, val = 1):
        self.__first = val
        ExampleClass.contador += 1
```

Una variable de clase **se inicializa** dentro de la clase pero fuera de cualquiera de sus métodos.

```
objeto_1 = ExampleClass()
objeto_2 = ExampleClass(2)

print(objeto_1.__dict__, objeto_1.contador)
print(objeto_2.__dict__, objeto_1.contador)
print(ExampleClass.__dict__, ExampleClass.contador)
```

```
{'_ExampleClass__first': 1} 2
{'_ExampleClass__first': 2} 2
{'__module__': '__main__', 'contador': 2, .....} 2
```

```
print(hasattr(ExampleClass, 'contador'))
```

True

Las variables de clase:

- ***no se muestran*** en el __dict__ de un objeto pero sí en el de la clase
- ***tienen el mismo valor*** en todas las instancias de clase (objetos)