

Documentación de la clase Society

Con esta clase definimos nuestra "sociedad", esta consta de un **nombre**, **características**, **especie**, **posición** en el terreno, y una instancia de **evolución**, además una variable para habilitar la misma. Evolution se explica en otro informe detalladamente.

```
In [ ]: import os
from sys import path
path.append(os.path.abspath(os.path.join('', os.pardir)))
from Simulation.society import Society
from Simulation.species import Species

humanidad = Species("humanidad")
society = Society("cubano", humanidad)

(society.characteristic, society.species, society.evolution, society.pos)

Out[ ]: ({'Poblacion': <Simulation.characteristic.Characteristic at 0x174707e9930>},
<Simulation.species.Species at 0x1746fd27cd0>,
<Simulation.evolution.Evolution at 0x174707e9960>,
None)
```

Características

Una sociedad posee una lista de **características** propias como pueden ser *población* (por defecto para todas las sociedades y especies), *economía*, *política*, *natalidad*, etc.

Para trabajar con las mismas se tiene los métodos:

- **Get_Characteristic_Value** : recibe el nombre de la característica y devuelve el valor de la misma
- **Change_Characteristic** : recibe todos los argumentos para crear una característica y la añade a la lista de características.
- **Delete_Characteristic** : recibe el nombre de la característica y la elimina de la lista de características.
- **Update_Characteristic_Value** : recibe nombre y nuevo valor, actualiza solo el valor de la características

```
In [ ]: society.Change_Characteristic("politica",1,0,10,3,lambda a: a*2)
society.Delete_Characteristic("politica")
society.Change_Characteristic("politica",1,0,10,3, None)
society.Update_Characteristic_Value("politica",2)
society.Get_Characteristic_Value("politica")

Out[ ]: 2
```

Además tiene un método **Copy** que devuelve una copia de la sociedad.