

Tutorial de Consola administrativa de Python

En este manual se le explicará cómo utilizar la consola de Python

Python es un lenguaje de programación **dinámico**, lo cual significa que el tipo de una referencia (y por ende la clase a la que apunta) puede variar al momento de ejecución, lo cual facilita la creación de un *REPL*, que es una consola que permite ejecutar scripts cortos que se insertan mediante un string o una terminal o un archivo como parte del funcionamiento de un programa. Python es usado en muchas áreas, pero es conocido principalmente por ser una muy versátil herramienta para procesamiento de datos, ya que posee la librería numpy para cálculos y transformaciones matriciales, pandas para interpretar y escribir datos en una inmensa cantidad de formatos, tensorflow para diseñar, ejecutar y entrenar redes neuronales artificiales, y una vasta librería de la comunidad así como una de las librerías estándar más amplia entre lenguajes de programación, posiblemente sólo vencida por la librería estándar de Common Lisp (o posiblemente ésta sea vencida por la de Python).

Al momento de la escritura de este documento, no se había incluido la librería estándar de Python en el intérprete aún, en caso de que esto cambie se agregará en la Errata del Sistema.

Ejemplo 1: Listar los autos en el Deposito de maldonado

Imagine que usted desea saber qué vehículos se encuentran actualmente en el Deposito de maldonado en el sistema. El código a ejecutar sería

```
f = Controladores.Fachada.getInstancia()
l = f.informacionBaseDelLugarPorNombre('Deposito de maldonado')
v = f.VehiculosLugar(l)
v
```

En la consola, usted verá la siguiente salida: `.net[1GH2J83LED0987547]`, lo cual significa que la función `VehiculosLugar` devolvió una lista del mundo `.net` que contiene un objeto de tipo *vehiculo* cuya representación en string es `"1GH2J83LED0987547"`.

```
f = Controladores.Fachada.getInstancia()
l = f.informacionBaseDelLugarPorNombre('Deposito de maldonado')
v = f.VehiculosLugar(l)
v
.net[1GH2J83LED0987547]
```

Suponga ahora que usted quiere recabar más información sobre dicho vehículo. Siendo Python un lenguaje dinámico, es imaginable que tenga una función que le permita inspeccionar fácilmente los miembros del objeto. Dicha función se llama `"dir"`. Ejecute en la consola el comando `dir(v[0])` y verá en la salida los miembros del objeto:

```
["AsTuple", "Año", "AñoNullable", "Cliente", "Color", "Equals", "GetHashCode", "GetType", "IdVehiculo", "Informes", "Marca", "MemberwiseClone", "Modelo", "ReferenceEquals", "TIPOS_VEHICULOS", "TIPO_VEHICULO_AUTO", "TIPO_VEHICULO_CAMION", "TIPO_VEHICULO_MINIVAN", "TIPO_VEHICULO_SUV", "TIPO_VEHICULO_VAN", "Tipo", "TipoBajaVehiculo", "TipoVehiculoIngresa", "ToString", "VIN", "__class__", "__delattr__", "__doc__", "__format__", "__getattr__", "__hash__", "__init__", "__new__", "__reduce__", "__reduce_ex__", "__repr__", "__setattr__", "__sizeof__", "__str__", "__subclasshook__"]
```

Algunos de ellos son parte de las funciones estándar de todo objeto en Python o en .Net, por

```
dir(v[0])
["AsTuple", "Año", "AñoNullable", "Cliente", "Color", "Equals", "GetHashCode", "GetType", "IdVehiculo", "Informes", "Marca", "MemberwiseClone", "Modelo", "ReferenceEquals", "TIPOS_VEHICULOS", "TIPO_VEHICULO_AUTO", "TIPO_VEHICULO_CAMION", "TIPO_VEHICULO_MINIVAN", "TIPO_VEHICULO_SUV", "TIPO_VEHICULO_VAN", "Tipo", "TipoBajaVehiculo", "TipoVehiculoIngresa", "ToString", "VIN", "__class__", "__delattr__", "__doc__", "__format__", "__getattr__", "__hash__", "__init__", "__new__", "__reduce__", "__reduce_ex__", "__repr__", "__setattr__", "__sizeof__", "__str__", "__subclasshook__"]
v[0].AsTuple()
(1GH2J83LED0987547, Fiat, Cronos, 2011, Sevel)
```

ejemplo `repr`, `hash`, `str`, `sizeof`, etc.

La función `"AsTuple"` le devolverá la información del vehículo en una tupla de python, ejecute el script `"v[0].AsTuple()"` y verá el resultado: `"(1GH2J83LED0987547, Fiat, Cronos, 2011, Sevel)"`.