# PYTHON

## IMPORTACIÓN DE MODULOS O CLASES

Para importar clases y poder usarlas en nuestro modulo lo podremos hacer de la siguiente forma:

* Importar clase concreta

from **[NOMBRE\_MODULO]** import **[NOMBRE\_CLASE]**

* + **Llamada: [NOMBRE\_CLASE].[FUNCION\_CLASE]()**
* Importar el módulo completo: de forma que se puede llamar a la clase

import **[NOMBRE\_MODULO]**

* + **Llamada: [NOMBRE\_MODULO].[NOMBRE\_CLASE].[FUNCION\_CLASE]()**
* Acceder a las funciones de un módulo mediante alias

from **[NOMBRE\_MODULO]** as **[NOMBRE\_MODULO]**

* + **Llamada: [NOMBRE\_CLASE].[FUNCION\_CLASE]()**

## CREACION DE OBJETOS

**Ejemplo**: Modulo *Coche.py*

class Coche:

‘ Constructor

def \_\_init\_\_(self, marca, modelo):

self.marca = marca

self.modelo = modelo

‘ Getters y Setters

def getMarca(self):

return self.marca

def setMarca(self, marca):

self.marca = marca

def getModelo(self):

return self.marca

def setModelo(self, modelo):

self.modelo = modelo

‘ Funciones de la clase

def info (self):

print ("Datos del coche:")

print ("Marca: " + str(self.marca))

print ("Modelo: " + str(self.modelo))

**Llamada:** *Modulo Principal.py*

from **[NOMBRE\_MODULO]** import **[NOMBRE\_CLASE]**

‘ Importamos la clase a utilizar

from **Coche** import **Coche**

‘ Generamos el objeto y llamamos a una de sus funciones

micoche = Coche (“Renault”, “Megane”)

micoche.**infoCoche()**

## LISTAS DE OBJETOS

**Ejemplo:**

def **autogen**():

listado = []

listado.append(Coche("Renault", "R24"))

listado.append(Coche("Ferrari", "F100"))

listado.append(Coche("Mclaren", "Mercedes"))

listado.append(Coche("Redbull", "Racing"))

return listado

‘ Generamos la lista de objetos usando el método creado

coches = **autogen**()

‘ Recorremos el listado y mostramos la información de cada coche usando el método info generado en la clase Coche

for coche in coches:

coche.**infoCoche**()

## LLAMADAS A METODOS Y FUNCIONES Y ENTRADA Y SALIDA DE DATO

En Python al igual que otros lenguajes es capaz de enviarse más de un parámetro a la función para tratar con ellos en ese bloque de código. Sin embargo, también es posible recibir más de un parámetro de salida de las mismas.

***- Función sin recibir datos de entrada ni de salida.***

# Función de prueba 01

def **func01**():

print (“Hola mundo”)

# Llamada

**func01**()

***- Función que recibe datos de entrada, pero no de salida.***

# Función de prueba 02

def **func02**(mensaje):

print (mensaje):

# Llamada

**func02**(“Hola mundo”)

***- Función que recibe datos de entrada, y devuelve de salida.***

# Función de prueba 03

def **func03**(num1, num2):

suma = num1 + num2

rest = num1 – num2

mult = num1 \* num2

divi = num1 / num2

return suma, rest, mult, divi

# Llamada

sum, res, mul, div **= func03**(10,100)