**FUNCIONES DE PRIMER ORDEN:**

Programación funcional ⇒ Funciones valores “normales”

* Caso 1 Genericidad → Las funciones pueden pasarse como parámetro.
* Caso 2 Instanciación → Las funciones pueden devolver otras funciones.
* Caso 3 Embebimiento → Las funciones pueden almacenarse en estructuras de datos.

**Caso 1. GENERICIDAD**: Las funciones se pueden pasar como parámetros

def saluda(nombre):

print "Hola %s" % nombre

def otrafuncion(funcion, parametro):

print "Soy funcion y ejecuto otra funcion"

funcion(parametro)

f=saluda // asignamos a f la funcion saluda

otrafuncion(f,"Pepe")

**Caso 2 INSTANCIACIÓN:** Las funciones pueden devolver otras funciones.

import random

def devuelvefun(param):

if param : **return random.random()**

else : return "hola"

fun=devuelvefun(True)

print fun

fun=devuelvefun(False)

print fun

**EMBEBIMIENTO** → Las funciones pueden almacenarse en estructuras de datos.

Nota: Importar la función del módulo y utilizarla directamente:

***from math import sin***

from operator import add,sub,div

funcs = {'+':add,'-':sub,'\*':lambda a,b:a\*b,'/':truediv }

print funcs ['+'] (1,2)

print funcs ['-'] (1,2)

print funcs ['\*'] (1,2)

print funcs ['/'] (1,2)

Otro uso con funciones propias

def sumar(a,b):

return a+b

def dividir(a,b):

return a/b

funcs = {'+’: sumar(1,2),'-':lambda a,b:a-b,'\*':lambda a,b:a\*b,'/':dividir(1,2) }

print funcs ['+']

print funcs ['-'] (1,2)

print funcs ['\*'] (1,2)

print funcs ['/']