Asier Marin

Laboratorio 2 – Introducción a Python

Esta actividad se entregará y realizara de manera individual, pero en el aula podéis consultar dudas y trabajar en la solución en parejas.

Entrega: este documento en formato Word o pdf, y todos los ficheros Python (.py) que hayáis creado.

**Actividad 1**

Crear en el intérprete una lista llamada valores mezclada de 2 enteros, 2 reales y 2 strings

• Mostrar por pantalla:

– Elemento 1

– Elementos del 0 al 3, incluidos

• Añadir un entero al final de la lista

• Modificar el elemento 1 asignando el valor “Casa”

• Volver a visualizar la lista

• Eliminar el elemento con subíndice 1 de la lista y volverla a visualizar

(código Python)

Dejo en enlace al repositorio donde están las actividades:

<https://github.com/asiermarin/Sistemas.Embebidos/blob/master/Python/Laboratorio.Introduccion.Python>

**Actividad 2 (diccionario.py):**

Crear un carpeta **labpython** y dentro de ella crear con nano un fichero diccionario1.py

• Crea un diccionario llamado persona con las claves nombre, apellido y edad, y unos valores cualquiera

• Visualiza la edad accediendo a su clave

• Salir de nano y ejecutar con las siguientes alternativas y comentar que ocurre y por qué.

python diccionario1.py

pyth on3 diccionario1.py

• Añadir la siguiente línea al principio del fichero

#!/usr/bin/env python

• Hacer el fichero ejecutable con chmod a+x diccionario1.py y ejecutar con

./diccionario1.py

• Añade al diccionario tras visualizar la edad, una nueva clave-valor llamada profesión

con el valor “ingeniero”

• Visualiza la persona

**Actividad 3** (no tenéis que entregar nada, solo es probar los comandos y observar que ocurre)

Probar en el intérprete

“Hola”[0:2]

“Hola”\*3

“o” in “Hola”

“Tu nombre es %s” % “John”

“Tu nombre es %s y tu apellido %s” % (“John”, “Rambo”)

“Tu nombre es %s y tu apellido %s, edad %d” % (“John”,

“Rambo”,32)

“La edad de %(nombre)s es %(edad)d” % {“edad”:34,

“nombre”:”Mikel”}

“La temperatura es %f“ % 23.4

“La temperatura es %.2f“ % 23.4

**Actividad 4**

Hacer un programa ejecutable en **Pyhton 3** denominado palabras.py

• Solicitar al usuario palabras en un bucle while y guardarlas en una lista hasta que escriba “fin”

• Posteriormente recorrer las palabras de la lista en un bucle for y por cada una mostrar por pantalla el número de letras que tiene

• La palabra de control “fin” no debe aparecer en la lista procesada

**Actividad 5 (calculadora/py)**

Diseñar en calculadora.py las funciones:

– sumar(op1,op2)

– restar(op1,op2)

– elevar(base,exponente)

• En la operación elevar(base,exponente) hacer que el parámetro exponente sea opcional, siendo el valor por defecto 2 en caso de no incluirlo.

• Invocar las diferentes funciones con o sin parámetros opcionales, mostrando el resultado por pantalla para validar el funcionamiento.

• EXTRA: Añadir al final del fichero un programa que muestre un menú con las operaciones (y una opción de salir), el usuario seleccione la deseada, luego introduzca los operandos, visualice el resultado y vuelva a mostrar el menú hasta que se seleccione la opción de salir.

**Actividad 6 (bisiesto.py)**

Escriba un programa que pida al usuario un año y que escriba si es bisiesto o no. Se recuerda que los años bisiestos son múltiplos de 4, pero los múltiplos de 100 no lo son, aunque los múltiplos de 400 sí. Estos son algunos ejemplos de posibles respuestas: 2012 es bisiesto, 2010 no es bisiesto, 2000 NO es bisiesto, 1900 no es bisiesto

**Actividad 7(sensor.py)**

• Crear un fichero sensor.py

• Definir una clase Medida con atributos “valor” y “unidad” que se reciben como

parámetros en el constructor.

• Definir una clase Sensor con un atributo llamado “tipo” que se recibe como parámetro

en el constructor y con otro atributo medidas que se inicializa como una lista vacía.

• Definir para cada una de las clases la función \_\_str\_\_(self) que permite invocar str(…)

sobre una instancia de la clase, retornando una representación en string de la misma.

En el caso del Sensor devolverá el tipo y el array de medidas.

• Hacer un programa que cree un sensor, le añada 3 medidas y al hacer

print(str(sensor)) o print(sensor) muestre, por ejemplo:

Sensor de temperatura

30 grados

29 grados

27 grados

**Actividad 7**

Usando la actividad 6, escribir un programa que cree y guarde los valores y el tipo del sensor.

Por ejemplo

(datosSensor.txt)

Sensor de temperatura

30 grados

29 grados

27 grados