Universidad de La Laguna
E.T. Ingeniería Civil e Industrial
Área: Lenguajes y Sistemas Informáticos

Programación Científica
Práctica de laboratorio **#1**

1. Iniciar una sesión de trabajo en GNU-Linux.

2. Muestre el árbol de directorios de su directorio HOME.

3. Cree un **Directorio de Proyecto** para la asignatura *Programación Científica* esto es en el directorio *PC* (`cd PC`).

4. Sitúese en su directorio de asignatura (`cd PC`).

5. Muestre el contenido del directorio de trabajo (`ls -la`).

6. Cree un nuevo directorio denominado *prct01* (`mkdir prct01`).

7. Sitúese en el directorio *prct01* (`cd prct01`) y cree la estructura de directorios que le permita tener subcarpetas para el código y los documentos, es decir:
   - un subdirectorio *src*
   - un subdirectorio *docs*

8. Guarde el fichero PDF que contiene el enunciado de esta práctica en el directorio *docs*.

9. En el directorio *docs* cree un fichero `respuestas.txt` en que almacene las respuestas a las preguntas de los ejercicios.

10. Sitúese en el directorio *src* y descomprima el fichero `prct01.tgz` que contiene los ficheros fuente que ponen de manifiesto la diferencia entre compilación e interpretación.
    (`tar -zxvf cvi.tgz`).

11. Compruebe que aparecen en su directorio actual los siguientes ficheros (`ls -la`).

```
helloWorld.c    - Lenguaje C   / compilado
helloWorld.cc   - Lenguaje C++ / compilado
helloWorld.sh   - Lenguaje Bash / interpretado
helloWorld.py   - Lenguaje Python / interpretado
HelloWorld.java - Lenguaje Java / compilado e interpretado
```

    Cada uno de los ficheros contiene un programa que muestra por pantalla la frase "Hello World".

12. Compilación en C:

    a) Muestre el contenido del fichero `helloWorld.c` sin abrirlo (`cat`).
    b) Compile el fichero `helloWorld.c` con el comando `gcc -o helloWorldC helloWorld.c`
    c) Compruebe que aparece en el directorio actual el fichero `helloWorldC` (`ls -la`)
    d) Ejecute el programa que se ha compilado con el comando `./helloWorldC`

13. Compilación en C++:

    a) Muestre el contenido del fichero `helloWorld.cc` sin abrirlo (`cat`)

    b) Compile el fichero `helloWorld.cc` con el comando `g++ -o helloWorldCPP helloWorld.cc`

    c) Compruebe que aparece en el directorio actual el fichero `helloWorldCPP` (`ls -la`)

    d) Ejecute el programa que se ha compilado con el comando `./helloWorldCPP`

14. Interpretación en Bash:

    a) Muestre el contenido del fichero `helloWorld.sh` sin abrirlo (`cat`)

    b) Ejecute el programa con el comando `bash ./helloWorld.sh`

15. Interpretación en Python:

    a) Muestre el contenido del fichero `helloWorld.py` sin abrirlo (`cat`)

    b) Ejecute el programa con el comando `python ./helloWorld.py`

16. Compilación e interpretación en Java:

    a) Muestre el contenido del fichero `HelloWorld.java` sin abrirlo (`cat`)

    b) Compile el fichero `HelloWorld.java` con el comando `javac HelloWorld.java`

    c) Compruebe que aparece en el directorio actual el fichero `HelloWorld.class` (`ls -la`)

    d) Ejecute el programa que se ha generado con el comando `java HelloWorld`

17. ¿Cuál es la diferencia entre compilación e interpretación?

18. ¿Qué permisos tiene el fichero `helloWorld.sh`? ¿Se puede ejecutar directamente? (`./helloWorld.sh`)

19. ¿Qué permisos tiene el fichero `helloWorld.py`? ¿Se puede ejecutar directamente? (`./helloWorld.py`)

20. ¿Cuál es la principal diferencia entre el contenido del fichero `helloWorld.sh` y `helloWorld.py`?

21. ¿En qué directorio está instalado el intérprete de Python? (`which python`)

22. Modifique el fichero `helloWorld.py` para que sea ejecutable. En primer lugar edite el fichero para añadir la primera línea correspondiente y a continuación modifique los permisos del fichero. (`chmod u+x helloWorld.py`)

23. Ejecutar el intérprete interactivo de Python. En este modo, se espera a la siguiente orden con el indicador principal, que suelen ser tres signos 'mayor' (`>>>`). Para las líneas adicionales, se utiliza el indicador secundario, que son tres puntos (`...`). El intérprete muestra un mensaje de bienvenida con su número de versión e información de derechos de copia, antes de mostrar el indicador principal. Para salir el intérprete interactivo se ha de teclear un carácter de final de fichero (`Control^+D`)

    ```
    $python
    Python 2.5.2 (r252:60911, Jan 20 2010, 21:48:48)
    [GCC 4.2.4 (Ubuntu 4.2.4-1ubuntu3)] on linux2
    Type "help", "copyright", "credits" or "license" for more information.
    >>>
    ```

24. Utilizar el intérprete interactivo de Python como una calculadora.

    ```
    >>> 2 + 2
    4
    ```

25. Escribir el programa “Hola Mundo” en Python usando el intérprete interactivo.

```
>>> hola = "Hola mundo desde Python"
>>> print hola
Hola mundo desde Python
```

26. ¿Qué sucede cuando se usa el operador de división entera (`//`) con números reales? Por ejemplo: (`1.0 // 3.0`)

27. ¿Qué hace el operador `**`? Por ejemplo: (`2 ** 3`)

28. ¿Qué hace el operador `%`? Por ejemplo: (`12 % 10`)

29. ¿Qué hacen los operadores `+=, -=, *=, /=, **=, //=`?

30. Evaluar las siguientes expresiones:

```
>> 2 + 3 < 2 * 3 or 6 < 10
?
>>> (2+3 < 2) * (3 or 6 < 10)
?
>>> a = 2 < 4
?
>>> b = 2 >= 4
?
>>> a * b
?
>>> True and True
?
>>> True and False
?
>>> False and True
?
>>> False and False
?
>>> True or True
?
>>> True or False
?
>>> False or True
?
>>> False or False
?
```

31. En Python hay números enteros y números reales. Los números enteros pertenecen a la clase denominada `int`, los números reales pertenecen a otra clase llamada `float`. `True` y `False` pertenecen a la clase `bool`. Para determinar la clase de un número o variable se usa la función `type()`. Comprobar su comportamiento con los siguientes ejercicios:

```
>>> type(10)
?
>>> type(10.0)
?
>>> type(False)
?
>>> type(True)
```

```
?
>>> type(10) is int
?
>>> type(10.0) is float
?
>>> type(10<20) is bool
?
>>> type(False) is bool
?
>>> type(False) is int
?
>>> type(10) == type(10.0)
?
>>> type(10+20) == type(10)
?
>>> type(10+20) == type(10<20)
?
```

32. Escriba varias cadenas (`strings`), asígnelas a variables y concaténalas entre sí. Por ejemplo:

```
>>> estrofa = " El patio de mi casa... "
>>> estribillo = " dubididu "
>>> estribillo += "zombie "*4
>>> cancion = estrofa + estribillo
```

33. Pruebe las siguientes expresiones de comparación entre cadenas:

```
>>> a = 'PYTHON'
>>> b = 'python'
>>> a == b
?
>>> a != b
?
>>> a != b
?
>>> a <= b
?
```

34. ¿Qué ocurre al concatenar objetos de tipo cadena con objetos numéricos? Por ejemplo:

```
>>> 'PYTHON' + 3
?
>>> 'PYTHON' + 3.1415
?
```

35. ¿Qué hace la función str()? Por ejemplo:

```
>>> 'PYTHON' + str(3)
?
>>> 'PYTHON' + str(3.1415)
?
```

36. ¿Qué resultado se obtiene al ejecutar las siguientes expresiones?

```
>>> "3" + 3
?
>>>int("3")+3
?
>>>"3"+str(3)
?
```

37. Se puede acceder a cada una de las letras de una cadena de caracteres usando el operador de indexación `[]`. ¿Qué resultados se obtienen al ejecutar lo siguiente?

```
>>> 'PYTHON'[0]
?
>>> cantante[0]
?
```

38. El mismo operador se utiliza para acceder a segmentos de la cadena (substrings). El primer número indica el índice de inicio y el segundo, separado por dos puntos, la longitud. ¿Qué resultados se obtienen al ejecutar?

```
>>> 'PYTHON'[0:3]
?
>>> cantante[2:4]
?
```

39. ¿Qué ocurre cuando se utilizan índices negativos para acceder a los elementos de una cadena? Por ejemplo: (`PYTHON[-1]`)

40. ¿Qué sucede cuando se omite parte del rango de una cadena? Por ejemplo, `PYTHON[3:] o PYTHON[:3]`

41. ¿Qué hace la función abs()? Por ejemplo:

```
>>> abs(-3)
?
>>> abs(3)
?
```

42. ¿Qué hace la función float()? Por ejemplo:

```
>>> float(3)
?
>>> float(3.2)
?
>>> float(3.2e10)
?
>>> float(Un Texto)
?
```

43. ¿Qué hace la función int()? Por ejemplo:

```
>>> int(2.1)
?
>>> int(-2.9)
?
>>> int('2')
?
```

44. ¿Qué hace la función round()? Por ejemplo:

```
>>> round(2.1)
?
>>> round(2.9)
?
>>> round(-2.9)
?
>>> round(2)
?
```

45. ¿Cuál es el resultado de evaluar las siguientes expresiones?

```
>>> abs(-23) % int(7.3)
?
>>> abs(round(-34.2765,1))
?
>>> str(int(12.3)) + 0
?
>>> str(float(str(2) * 3 + .123)) + 321
?
>>> str(int(2.1) + float(3))
?
```

46. ¿Cuál es el resultado de ejecutar las siguientes sentencias?

```
>>> from math import sin
>>> sin(0)
?
>>> sin(1)
?
>>> cos(0)
?
>>> from math import cos
>>> cos(0)
?
>>> from math import *
>>> sin(2*pi)
?
```

47. ¿Cuál es el resultado de evaluar las siguientes expresiones?

```
>>> from math import *
>>> int(exp(2 * log(3)))
?
>>> round(4*sin(3 * pi / 2))
?
>>> abs(log10(.01) * sqrt(25))
?
>>> round(3.21123 * log10(1000), 3)
?
```

48. ¿Cuál es el resultado de ejecutar las siguientes sentencias?

```
>>> print "%d" %1
?
>>> print "%d %d" %(1, 2)
?
>>> print "%d%d" %(1, 2)
?
>>> print "%d, %d" %(1, 2)
?
>>> print 1, 2
?
>>> print "%d 2" %1
?
```

49. Escriba un programa Python, en el entorno interactivo, que calcule el promedio de las calificaciones de tres asignaturas en tres acumuladores independientes y luego calcule el promedio general de todas.

50. Situado en el directorio de la asignatura, es decir, en el directorio `\PC` comprima las actividades de la práctica (`tar -zcvf prct01.tgz prct01/`).

51. Compruebe que se ha creado el fichero `prct01.tgz` correctamente en el directorio actual (`tar -ztvf prct01.tgz *`).

52. Suba el fichero `prct01.tgz` a la tarea habilitada en el campus virtual.

53. Cierre la sesión.