

1. Iniciar una sesión de trabajo en GNU-Linux.
2. Muestre el árbol de directorios de su directorio HOME.
3. Sitúese en el directorio de la asignatura “Programación Científica”(cd PC).
4. Muestre el contenido del directorio de trabajo (ls -la).
5. Cree un nuevo directorio denominado *prct03* (mkdir prct03).
6. Sitúese en el directorio *prct03* (cd prct03) 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*
7. Guarde el fichero PDF que contiene el enunciado de esta práctica en el directorio *docs*.
8. En el directorio *docs* cree un fichero **respuestas.txt** en que almacene las respuestas a las preguntas de los ejercicios.
9. En el directorio *src* cree un subdirectorio **ejemplo**.
10. Descargue el fichero **mp.tgz** en el directorio *src/ejemplo* y descomprálos. (tar -zxvf mp.tgz)
11. Sitúese en el directorio *src/ejemplo* y compruebe que aparecen los siguientes ficheros (ls -la).

gravedad.py	# Modulo Python con constantes y fuciones de física gravitatoria
main_gravedad.py	# Ejemplo de inclusión de pruebas en los módulos Python
uso_gravedad.py	# Ejemplo de uso del módulo Python gravedad
12. Muestre el contenido del fichero **gravedad.py**. (cat gravedad.py)
13. Describa el contenido del fichero **gravedad.py**. ¿Cuántas partes se distinguen? ¿Qué clase de elementos aparecen en el mismo?
14. ¿Qué es un *módulo*?
15. Muestre el contenido del fichero **main_gravedad.py**. (cat main_gravedad.py)
16. Describa el contenido del fichero **main_gravedad.py**. ¿Cuál es la principal diferencia con el fichero **gravedad.py**?

17. Ejecute el fichero `main_gravedad.py`. (`python main_gravedad.py`)
18. Describa la salida de la ejecución del fichero `main_gravedad.py`. ¿De qué clase de elementos se está verificando el correcto funcionamiento?
19. Muestre el contenido del fichero `uso_gravedad.py`. (`cat uso_gravedad.py`)
20. Describa el contenido del fichero `uso_gravedad.py`. ¿Cuál es la principal diferencia con el fichero `main_gravedad.py`?
21. Ejecute el fichero `uso_gravedad.py`. (`python uso_gravedad.py`)
22. Compruebe que aparecen en su directorio actual el fichero `gravedad.pyc` (`ls -la`).
23. Describa el contenido del fichero `gravedad.pyc`.
24. ¿En qué directorio está instalado el intérprete de Python? (`which python`)
25. Modifique el fichero `uso_gravedad.py` para que sea ejecutable. (añadir en la primera línea del fichero `uso_gravedad.py` lo siguiente `#!` < resultado de la pregunta 24 >)
26. Ejecute el fichero `uso_gravedad.py`. (`./uso_gravedad.py`)
27. En caso necesario modifique el fichero `uso_gravedad.py` para que sea ejecutable. (`chmod u+x uso_gravedad.py`)
28. El objetivo de los siguientes ejercicios es crear un módulo escrito en Python. Con el programa se comprobará el mantenimiento de las identidades matemáticas en una máquina de cómputo. Cree un subdirectorio llamado *identidades* dentro de *src/*.
29. Debido a los errores de redondeo, puede suceder que una regla matemática como $(ab)^3 = a^3b^3$ no se mantenga en un ordenador.

El objetivo de este ejercicio es generar una gran cantidad de números de forma aleatoria y comprobar con ellos tales identidades.

Los números aleatorios se generan en Python utilizando el módulo Random:

```
import random
a = random.uniform(A, B)
b = random.uniform(A, B)
```

En este caso, a y b serán números aleatorios mayores o iguales que A y menores que B .

Escriba un programa que reciba como entrada el número de test que se van a realizar desde la línea de comandos.

En Python la lista `sys.argv[1:]` contiene todos los argumentos de la línea de comandos que se le pasan al programa. Todos los elementos en `sys.argv` son cadenas. `sys.argv[0]` contiene el nombre del programa. Una forma general de uso es la siguiente:

```
parametro1 = float(sys.argv[1])
parametro2 = float(sys.argv[2])
parametro3 = sys.argv[3]
```

Establecer los valores de A y B para que sean fijos (por ejemplo de -100 a 100).

Realizar la prueba en un bucle. Dentro del bucle, generar dos números aleatorios a y b y comprobar si las dos expresiones matemáticas $(a * b) * *3$ y $a * *3 * b * *3$ son equivalentes.

Contar el número de fallos de equivalencia y mostrar como salida el porcentaje de fallos.

30. Situado en el directorio de la asignatura, es decir, en el directorio `PC/` comprima las actividades de la práctica (`tar -zcvf prct03.tgz prct03/`).
31. Compruebe que se ha creado el fichero `prct03.tgz` correctamente en el directorio actual (`tar -ztvf prct03.tgz *`).
32. Suba el fichero `prct03.tgz` a la tarea habilitada en el campus virtual.
33. Cierre la sesión.