

# Ejercicios de Python I

## ***El modo interactivo.***

Para entrar en el modo interactivo de *Python* ejecutar la orden `python` en una terminal. Hacer los siguientes ejercicios:

1. Comprobar qué versión del Python estamos utilizando.
2. Probar el modo calculadora. Operaciones con `+`, `-`, `*`, `/`, `**`, `%`, `(`, `)`, etc.
3. Probar la definición de variables de varios tipos (`int`, `float`, `string`) y usarlas posteriormente en las operaciones del apartado anterior.
4. Probar la función `print` con las variables anteriores y con literales.
5. Usar la coma para separar varios elementos en un mismo `print`. Probar `print("hola"); print("y adiós")`. Observar la coma detrás del primer `print`.
6. Probar `print` con cadenas, `bool`, `int` y `floats`.
7. Importar el módulo `math` y mostrar las funciones que incorpora
8. Usar algunas de las funciones del módulo `math`.
9. Intentar salir de Python con `quit` y con `CTRL+D`.
10. El modo interactivo es muy útil en Python y es importante desenvolverse bien en él.

## ***La ayuda del modo interactivo:***

1. Para pedir la ayuda de una palabra clave en Python se pone entre paréntesis y la palabra clave entre comillas. Para pedir ayuda de la orden `import` escribimos: `help("import")`. Probar esto con `import`, `print`, etc.
2. Las instrucciones:  

```
import rlcompleter, readline
readline.parse_and_bind("tab:complete")
```

permiten completar instrucciones con ayuda del tabulador. Por ejemplo, después de importar el módulo `math`, escribir `math.` y pulsar dos veces el tabulador. Comprobar que salen todas las funciones de `math`.
3. De igual forma invocar con `help` la ayuda de algunas de las funciones del módulo `math`.
4. Entrar en el sistema de ayuda interactiva con `help()`. Ahora no es necesario escribir `help()` cada vez que se solicita ayuda, se pone la palabra clave o el tópico directamente y sale la ayuda.
5. Para ver una lista con los temas de ayuda ejecutar la orden `topics`.
6. Solicitar ayuda de alguno de los *topics* anteriores.
7. El sistema de ayuda es muy importante en Python, y es muy importante y conveniente familiarizarse bien con él. Nos sacará de muchos aprietos.

## ***El editor idle***

1. Entrar dentro ejecutando la orden `idle`
2. Crear el fichero `hola.py` con el típico programa que visualice la cadena "hola Mundo".
3. Guardar el fichero y ejecutar nuestro primer programa en python desde la shell de UNIX. Ojo con los permisos de ejecución y la ruta hasta python.
4. Ejecutar el programa anterior desde dentro de IDLE con el menú 'Run',

- opción 'Run Module' o bien F5.
5. Navegar los menús de IDLE y comprobar su utilidad. Este editor puede usarse con Python, pero también cualquier otro editor de código; hacer la prueba con cualquier otro editor (nedit, gedit, etc).

## **Funciones**

1. Crear un fichero `funciones.py` que contenga una función que se llame `suma` y que devuelva la suma de los dos parámetros y un programa principal que llame a la función anterior con dos valores cualesquiera.
2. Ejecutar el programa desde la *shell* del SO.
3. Entrar en el modo interactivo e importar el módulo `funciones`, y sin salir del modo interactivo invocar a la función `suma` con enteros, reales y cadenas. Observar que después de importar el módulo `funciones`, para invocar a la función `suma` sería: `funciones.suma()`.
4. Añadirle una cadena de documentación a la función `suma`, volver a importar el módulo `funciones` e invocar la cadena de documentación de la función `suma` que acabamos de añadir. ¡Ojo!: para recargar un módulo una vez modificado usar: `reload(modulo)`.
5. Hacer otras funciones (resta, multiplicación, etc.) probando y repitiendo lo anterior.

## **Cadenas**

1. En el modo interactivo, crear un par de variables con sendas cadenas y construir una cadena nueva a partir las otras dos anteriores mediante concatenación.
2. Usar el *slicing* de cadenas para obtener: el principio de una cadena hasta cierta posición, el final de una cadena desde cierta posición, una subcadena desde una posición a otra, etc.
3. Intentar acceder a una posición no existente de una cadena.
4. Modificar el valor de una variable de tipo cadena previamente creada.
5. Comprobar la utilidad de `upper`, `lower`, `strip`, `max` (con una y varias cadenas), `min` (con una y varias cadenas), `n`, `not in`, etc.
6. Importar el módulo `string`. Usar: `letters`, `lowercase`, `uppercase`, `digits`, `punctuation`, y el resto de utilidades. Acudir a la ayuda para aprender a usarlas.
7. Importar el módulo `string`. Ver la ayuda de la función `string.split()`. mediante la orden `help(string.split)` y probarla en el modo interactivo. Hacer lo mismo con `find`, `replace`, `join`.
8. Pedir al usuario una frase y mostrar en pantalla el número de palabras de esa frase.