

# Python (Ejercicios)

## Objetivo

Se trata de repasar todos los conceptos de programación en Python necesarios para poder pasar a la siguiente fase de desarrollo. Alguno de los elementos que te encontrarás en este cuestionario los habrás ya empleado en tus prácticas, otros no. Recuerda que la inspiración no llega de forma pasiva, así que recurre a todas las fuentes de información que necesites; libros, wiki, profesor... En cualquier caso, suerte y buen trabajo.

## Procedimiento

- Crea un documento de texto en Google Docs y copia en él las preguntas que encontrarás más abajo. Alternativamente, hazlo en un documento de Open Office y súbelo a Google Docs.
- Compártelo (solo) con tu profesor para que pueda seguir tus progresos y corregirlo cuando lo acabes (menú 'Compartir' -> 'Invitar a usuarios' y escribe su e-mail; asegúrate que está marcado 'Para editar')
- Responde cada pregunta escribiendo las respuestas inmediatamente a continuación. Hazlo en color azul para que se distinga con facilidad. Escribe todo el código conservando con fidelidad todos los sangrados que, como sabes, son tan importantes en Python. Añade todas las explicaciones que necesites.
- Cuando lo tengas acabado, renombra el archivo y añade la palabra 'Finalizado' (menú 'Archivo' -> 'Cambiar nombre...')

## Ejercicios

1. Escribe un ejemplo de cada tipo de dato que conoces de Python (es decir, un entero, un decimal, una cadena de texto...) indicando cada tipo con su nombre pythónico.
2. Convierte el entero 3 en un float y en una string. ¿Cómo convertirías el texto 'hola' en una lista? ¿Qué obtienes?
3. Calcula el resto y el cociente de la división entera entre 3.2 y 1.1
4. Investiga: Cuando en el intérprete escribes 3.5, éste devuelve 3.5. Pero cuando escribes 3.6 ¿qué obtienes? ¿Por qué? Como puedes imaginar, en las operaciones numéricas más habituales no influye mucho, pero si se desean cálculos de altísima precisión, puede remediarse. ¿Cómo?
5. Puedes convertir fácilmente una lista en una tupla y viceversa. Pero, ¿qué diferencia hay entre ellas? ¿Puedes poner algún ejemplo que lo ilustre?

6. Dada la lista [3,-2,5,7,0,1.5] Extrae con una instrucción de Python la sublista [5,7,0]. Añádele el número 3.14. Elimina el 7.
7. Dada la tupla (3,6,9,10, 17) extrae con una instrucción python la subtupla (3,9,17). ¿Sabrías obtener la tupla invertida?
8. Escribe una función que, dada una palabra, devuelva la primera letra.
9. Escribe una función que, dada una palabra diga si empieza por mayúscula (es decir, devuelva True en ese caso y False en caso contrario).
10. Escribe una función que, dado un texto, devuelva el número de palabras que contiene.
11. Escribe una función que, dado un texto, lo escriba invertido.
12. Escribe una función que devuelva una lista con los cuadrados de los números del 1 al 10.
13. Se trata ahora de hacer lo anterior de otra forma. ¿Sabrías hacerlo en una sola línea de código usando listas por comprensión?
14. ¿Y si se te pide, en lugar de los cuadrados, los cubos de los números impares entre 1 y 10?
15. Escribe una función que, dado un texto, diga si es una dirección de correo electrónico válida (es decir, debería devolver True si es algo del tipo [texto@texto.texto](#) y False en caso contrario).
16. Construye un diccionario con 6 nombres y sus números de teléfono. Escribe una función que, dado un nombre, te devuelva su número de teléfono. Escribe otra función que dado un número de teléfono, diga a quién pertenece. Por supuesto, en ambos casos, si no existe el programa no debe de dar error si no que amablemente debe decirlo. Reúne todo esto en un programa que permita a su usuario utilizar el diccionario o dejar de hacerlo, a su elección.