

# EJERCICIOS DE INTRODUCCIÓN A PYTHON

## PARTE IV

## INTRODUCCIÓN

---

El propósito de esta práctica es afianzar los conocimientos que habéis adquirido durante la asignatura: "Introducción a Python".

Este ejercicio es auto corregible. Para ver el archivo adjunto que contiene la solución propuesta por los docentes, únicamente deberás marcar la sección dónde se encuentra el ejercicio como finalizado.

Al pulsar sobre la opción:



Imagen que muestra el botón que permite marcar una sección para terminar en el campus

Se actualizará la sección y te aparecerá no sólo el archivo con el enunciado si no también el archivo con la solución. Ejemplo:



Enunciado  
M052\_CP\_04\_01\_E.pdf  
**DESCARGAR**



Solución  
M052\_CP\_04\_01\_S.php  
**DESCARGAR**

Imagen que muestra cómo se mostraría la solución

## Instrucciones para realizar los ejercicios

---

Intenta no ayudarte del temario para la realización de estos ejercicios. Si finalmente necesitas repasarlo para terminarlos te recomiendo que los repitas cada día hasta que ya no te haga falta utilizarlo. Así interiorizarás las bases de Python, así como su sintaxis.

Si tuvieses problemas para realizarlos incluso con la ayuda del temario no dudes en ponerte en contacto con los docentes del curso a través la mensajería del campus.

Desde el equipo docente te recomendamos que para la realización de esta práctica crees una carpeta en una ubicación determinada de tu ordenador. En ella deberás crear un archivo con la extensión .py (archivo de Python) o .ipynb (archivo de Jupyter Notebook) para cada ejercicio propuesto en este enunciado. Puedes hacer uso del programa que te resulte más cómodo.

**Ejercicio 1\_** Crea un programa que:

- Defina una función que reciba tres números enteros como parámetros y los muestre por pantalla.
- Solicite al usuario que introduzca tres números enteros
- Ejecute la función definida anteriormente pasándole como parámetro los tres números recibidos del usuario.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 2\_** Crea un programa que:

- Defina una función que permita calcular la suma de los valores de una lista de números que deberá recibir como parámetro. La función deberá mostrar el resultado por pantalla.
- Declare una lista de números enteros (pueden ser aleatorios)
- Ejecute la función definida anteriormente pasándole como parámetro la lista de números enteros creada.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 2.Alternativo \_** Crea un programa que:

- Defina una función que permita calcular la suma de los valores de una lista de números que deberá recibir como parámetro. La función deberá devolver el resultado.
- Declare una lista de números enteros (pueden ser aleatorios).
- Declare una variable dónde asociar el resultado de la ejecución de la función definida anteriormente. Luego se deberá mostrar el valor de dicha variable por pantalla.

**Ejercicio 3 \_**Crea un programa que:

- Defina una función que reciba una lista de números enteros como parámetro y compruebe qué números de la lista son pares y los muestre por pantalla.
- Declare una lista de números enteros (pueden ser aleatorios)
- Ejecute la función previamente definida pasándole como parámetro la lista de números enteros que se ha declarado.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 3.Alternativo \_**Crea un programa que:

- Defina una función que reciba una lista de números enteros como parámetro y compruebe qué números de la lista son pares, introduciéndolos en una nueva lista que se devolverá como resultado de la función.
- Declare una lista de números enteros (pueden ser aleatorios)
- Declares una variable dónde asociar el resultado de la ejecución de la función definida anteriormente. Luego se deberá mostrar el valor de dicha variable por pantalla.

**Ejercicio 4 \_**Crea un programa que:

- Defina una función que realice la suma de todos los números comprendidos entre 0 y 10 (ambos inclusive) y devuelva la suma como resultado de la función.
- Declare una variable dónde asociar el resultado de la ejecución de la función definida anteriormente. Tras ello, se deberá mostrar el resultado por pantalla.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 5** \_ Crea un programa que:

- Defina una función que se encargue de realizar una pregunta sencilla al usuario y evaluar su respuesta.

Si la respuesta proporcionada por el usuario es afirmativa (Si o similar) la función deberá mostrar un mensaje por pantalla.

Si la respuesta proporcionada por el usuario es negativa (No o similar) la función deberá mostrar un mensaje diferente por pantalla.

Si la respuesta proporcionada por el usuario es diferente a la esperada (no es ni positiva ni negativa) entonces se le deberá mostrar un mensaje de error indicando que no se ha comprendido su respuesta y se deberá volver a repetir el código interno de la función, de forma que se vuelva a realizar la pregunta y solicitar una respuesta.

- Ejecute la función previamente definida.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 5.Alternativo \_** Crea un programa que:

- Consulte con el usuario cuantas preguntas y acciones desea registrar, comprobando que la cantidad introducida por el usuario sea válida.
- Consulte con el usuario qué preguntas y acciones desea registrar. La cantidad de preguntas/acciones a registrar deberá coincidir con la cantidad solicitada en el paso anterior.
- Consulte tras introducir cada pregunta/acción si es correcta o desea modificarla. En caso de que lo sea se continuará con el siguiente apartado y en caso de que no, se le redirigirá al paso anterior.
- Registre cada pregunta/acción validada en el paso anterior dentro de un mismo diccionario.

Tras los pasos anteriores, el programa deberá:

- Informar de la cantidad de preguntas registradas en el diccionario y preguntar si desea responder alguna de ellas. Si solo hubiese una única pregunta, el mensaje le indicará si desea responder la pregunta existente. En caso de que haya varias le tendrá que preguntar cuántas desea responder. En esta última situación el programa deberá elegir preguntas aleatorias del diccionario en función de la cantidad indicada por el usuario.
- El programa deberá comprobar que la cantidad de preguntas que desea responder el usuario sea válida y en caso contrario volver a preguntarle.
- El programa deberá plantear las preguntas al usuario y en función de su respuesta enviarle un mensaje acorde, de la misma forma que se indicó en el ejercicio 5.

*Ejemplo de funcionamiento:*

Programa: ¿Cuántas preguntas desea registrar?

Usuario: 1

Programa: Introduzca la pregunta que desea registrar

Usuario: ¿Desea apagar el equipo?

Programa: Introduzca la acción asociada a la pregunta registrada

Usuario: "apagado del equipo"

Programa: La información proporcionada es .... ¿Es correcta? Responda SI o NO

Usuario: SI

Programa: Se ha registrado su pregunta

Programa: Hay una pregunta registrada ¿Desea responderla? Responda SI o NO

Usuario: SI

Programa: ¿Desea apagar el equipo?

Usuario: NO

Programa: Respuesta aceptada. No se procederá con "apagado del equipo".

**Ejercicio 6** \_ Crea un programa que:

- Defina una función que reciba como parámetro una lista o una tupla de números enteros. La función deberá comprobar cuantos números de la lista tienen un valor inferior a 10.
- Defina una tupla o lista de números enteros.
- Ejecute la función pasándole la lista definida como parámetro y mostrando el resultado por pantalla.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 6.Alternativo** \_ Crea un programa que:

- Defina una función que devuelva como resultado una lista de 20 números enteros aleatorios entre 0 y 100 (ambos inclusive).
- Defina una función que reciba como parámetro una lista de números. La función deberá registrar cuantos números inferiores a 10 (sin repetir) hay en la lista y cuáles son. La función deberá devolver estos dos datos como resultado.
- Defina una variable dónde asociar la lista de números aleatoria resultado de la ejecución de la primera función.
- Defina dos variables dónde asociar los resultados de la ejecución de la segunda función.
- Muestre por pantalla el resultado obtenido.

*Ejemplo de funcionamiento:*

**Lista de números:** 3, 30, 3, 3, 43, 35, 94, 94, 75, 50, 21, 10, 50, 58, 80, 84, 2, 2, 2, 3

**Resultado:** Existen un total de 2 números únicos en la lista inferiores a 10, que son: [2, 3]

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Alternativa2" que se encarga de mostrar los valores repetidos y no los únicos como en el ejercicio llamado "Alternativo"*

**Ejercicio 7** \_ Crea un programa que:

- Defina una función que te permita calcular el sueldo semanal de un trabajador y mostrarlo por pantalla. La función deberá recibir como parámetros: el número de horas trabajadas y el importe que gana el trabajador por cada hora trabajada.

Sabiendo que:

- La jornada normal (semanal) se considera de 35h.
  - Las horas extras trabajadas se pagan 10€ más que la hora normal.
- Solicite al usuario la cantidad de horas que ha trabajado esa semana y el precio al que le pagan la hora normal.
  - Ejecute la función pasándole los valores necesarios como parámetros.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada “Mejorado” con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio 8** \_ Crea un programa que permita calcular un presupuesto de viaje, sabiendo que:

- Los destinos disponibles para calcular el presupuesto serán: Alicante, Barcelona y Sevilla
- El coste del viaje a Alicante será de 250€, a Barcelona de 450€ y a Sevilla de 400€
- El precio de la noche de hotel será de 140€ independientemente del destino.
- El precio del alquiler de los vehículos en la ciudad destino será de 35€/día, independientemente del destino. Si el cliente desea reservar el vehículo un mínimo de 3 días se le aplicará un descuento al precio total del alquiler del 5%. Si lo reserva un mínimo de 7 días el descuento sobre el precio total del alquiler del vehículo será del 20%.

Para la realización de este ejercicio deberás consultar con el usuario a qué destino desea viajar, cuantas noches se alojará en un hotel (si quiere alojarse en uno) y cuantos días alquilará un vehículo (si desea alquilar alguno).

Deberás intentar realizar este ejercicio aplicando todos los conocimientos que has aprendido hasta el momento. Se recomienda que crees como mínimo una función independiente para:

- Calcular el coste del viaje a la ciudad destino
- Calcular el coste del hotel
- Calcular el coste del alquiler del vehículo
- Calcular el coste total del viaje. Esta función además deberá mostrar por pantalla el resultado obtenido.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*

**Ejercicio Extra \_** Crea un programa que:

- Defina una función que reciba dos números enteros como parámetros y calcule cuál de los dos es mayor, devolviendo el dato como resultado de la función.
- Defina una función que reciba tres números enteros como parámetros y calcule cuál de los tres es mayor, haciendo uso de la función definida previamente.
- Solicite al usuario que introduzca tres números enteros
- Ejecute la segunda función que se ha definido anteriormente y se le facilite como parámetro los tres números recibidos del usuario.

*Nota: En la carpeta de soluciones se adjunta una solución adicional llamada "Mejorado" con varias mejoras. Esta solución puede implementar funciones adicionales en el código.*