

30 Ejercicios de Python

Juan Monroy



¡Hola! Soy Juan, gracias por descargar este documento.

Quiere decir que tienes un compromiso con tu aprendizaje. ¡Enhorabuena!

Te planteo, nada más y nada menos que, 30 ejercicios de Python. 30.

Y te reto a hacerlos. Todos. Toditos. Por ejemplo, uno al día durante un mes.

¿¡Cómo!? ¿Que no están resueltos?

¡Exacto! No quiero darte la solución¹, quiero darte el problema. Quiero que te pelees con ellos. Quiero que avances. Quiero que aprendas.

De hecho, lo importante no es la solución. Lo que de verdad importa es que te enfrentes a ellos y los resuelvas, mejor o peor, pero que los resuelvas. La práctica es lo que importa.

Los primeros son muy fáciles pero van incrementándose en dificultad poco a poco.

¿Eres capaz de resolverlos todos?

¡Adelante!

¹ ¡Eh! Por aquí... Shhh. Pequeña actualización: finalmente, muchos meses después, he resuelto los ejercicios y los he puesto a disposición de quien los necesite, en mi academia, por un precio ridículo. Más información al final de este documento. Pero, de momento, olvídate de esto y céntrate en resolver los ejercicios.

Ejercicios:

1. Nada de "Hola, Mundo"	3
2. Numeritos, numeritos.....	3
3. Máximos y mínimos, pares e impares.....	4
4. Listas y listillos.....	4
5. Porciones.....	4
6. Un poco de geometría.....	5
7. Convirtiendo cosas.....	5
8. Devolviendo el cambio.....	5
9. Matematicando.....	5
10. Sucesiones típicas.....	6
11. Dibujitos.....	6
12. Passwords.....	7
13. Un día extra.....	7
14. Agenda.....	7
15. Evitando duplicados.....	8
16. Ordenando cositas.....	8
17. Sudokus.....	8
18. Matricitas.....	9
19. Contando valores.....	9
20. Suma de los números de una frase.....	9
21. Tecleando.....	9
22. Aplanando listas.....	10
23. Ordenación rápida.....	10
24. Buenos pares.....	10
25. Números romanos.....	10
26. A pedradas.....	11
27. Bombones.....	11
28. Detectando transacciones inválidas.....	12
29. Eliminando dígitos.....	12
30. Caramelos.....	12
Comentarios finales.....	13
Actualización e información IMPORTANTE.....	14

1. Nada de “Hola, Mundo”

Le tengo un poco de manía al típico ejercicio de “Hola, Mundo”. Esta es una alternativa un pelín más compleja.

Haz un programa en Python que pida por teclado al usuario su nombre, su edad y su altura en metros.

Después muestra por pantalla un mensaje del tipo:

“El usuario NOMBRE, de EDAD años de edad y mide ALTURA metros”.

¿Eres capaz de modificarlo para que el mensaje sea de la siguiente manera? Solo pidiendo los tres valores indicados.

“El usuario NOMBRE, de EDAD años de edad y mide METROS metros y CENTÍMETROS centímetros.”.

Ejemplos:

“El usuario Juan tiene 44 años de edad y mide 1.85 metros”.

“El usuario Juan tiene 44 años de edad y mide 1 metro y 85 centímetros”.

2. Numeritos, numeritos

Haz un programa de Python que pida un número entero N al usuario entre 10 y 20.

Muestra por pantalla todos los números enteros entre el 1 y el N.

Muestra por pantalla todos los números enteros entre el 30 y el N, en orden inverso.

3. Máximos y mínimos, pares e impares

Haz un programa que pida tres números al usuario y muestre por pantalla cuál es el mínimo y cuál es el máximo y que indique si son pares o impares. No uses listas.

4. Listas y listillos

Haz un programa en Python que pida 10 números al usuario y los almacene en una lista.

Muestra todos los números por pantalla indicando junto a cada número un mensaje que diga "NEGATIVO" si dicho número es negativo.

¿Lo tienes? Modifícalo para que muestre "REPETIDO" al lado de cada número que se encuentre más de una vez en la lista.

5. Porciones

Haz un programa en Python que pida números al usuario que se vayan almacenando en una lista. El programa deja de pedir números cuando el usuario introduce un cero.

Muestra por pantalla los números en posición par de la lista.

Muestra por pantalla los números en orden inverso.

Parte la lista en dos mitades. Si el número de elementos es impar, no importa qué mitad tiene un elemento más. Muestra las dos mitades por pantalla.

Muestra por pantalla todos los elementos de la primera mitad, exceptuando el primero y el último. Muestra por pantalla el máximo y el mínimo de la segunda mitad.

6. Un poco de geometría

Pide al usuario las coordenadas X e Y de dos puntos en el espacio.

Muestra por pantalla la distancia en línea recta entre esos dos puntos.

Muestra por pantalla las coordenadas del punto medio exacto de ambos.

7. Convirtiendo cosas

Haz un programa en Python para convertir de un número decimal a uno binario.

Y para convertir uno binario en decimal.

No te olvides del octal y el hexadecimal.

8. Devolviendo el cambio

Haz un programa que reciba un número que represente una cantidad de dinero D, incluyendo céntimos.

Calcula la mínima cantidad de monedas necesarias para sumar D, suponiendo que tenemos monedas de 2, 1, 0.5, 0.2, 0.1, 0.05, 0.02, y 0.01.

Muestra por pantalla la cantidad de cada una de las monedas.

9. Matematicando

Haz un programa en Python que pida dos números enteros N y M al usuario.

Calcula y muestra por pantalla el máximo común divisor (MCD) de N y M.

Calcula y muestra por pantalla el mínimo común múltiplo (MCM) de N y M.

¿Lo tienes? Ahora haz lo mismo pero pidiendo varios números al usuario. Almacénalos en una lista y calcula el MCM y el MCD de todos ellos.

10. Sucesiones típicas

Haz un programa que muestre por pantalla los N primeros elementos de la sucesión de Fibonacci.

Mmmm, y ahora, escríbelos a un fichero llamado fibonacci.txt.

11. Dibujitos

Haz un programa en Python que dibuje por pantalla los siguientes patrones para un número N leído por teclado. Por ejemplo, para N = 5:

```
* * * * *
* * * *
* * *
* *
*
```

```
* * * * *
*       *
*       *
*       *
* * * * *
```

```
* * *  
 * *  
* * *  
 * *  
* * *
```

```
 *  
***  
*****
```

12. Passwords

Haz un programa en Python que compruebe la validez de un password introducido por pantalla.

El password debe cumplir todas estas condiciones para ser válido:

- Al menos una letra minúscula (a-z) y al menos una mayúscula (A-Z).
- Al menos un número (0-9).
- Al menos un símbolo (#, \$, /, @).
- Al menos 10 caracteres de longitud.

13. Un día extra

Haz un programa que muestre por pantalla todos los años bisiestos entre el 1 y N para cualquier número entero N introducido por teclado.

14. Agenda

Tenemos un fichero llamado agenda.txt que contiene contactos telefónicos. Cada línea está compuesta de un número de identificación único, un nombre (pueden ser varias palabras) y un número de teléfono.

Por ejemplo, el contenido de un fichero con dos líneas puede ser el siguiente:

```
20 María López +34612000000
30 Pepe Martínez Contreras +17610111111
```

Haz un programa que lea un fichero con este formato y cargue los datos en memoria. A continuación pedirá al usuario un número de identificación por teclado y el programa debe mostrar los datos del contacto con dicho número de identificación, en este formato (para el identificador 20):

```
Nombre: María Pérez
Teléfono: +34612000000
```

15. Evitando duplicados

Haz una función en Python que reciba una lista por parámetro y devuelva dos listas diferentes, una con los elementos de la lista recibida, pero sin los repetidos, y otra con los repetidos.

Por ejemplo, de [1, 2, 3, 2, 2, 3, 4] se devuelve [1, 2, 3, 4] y [2, 2, 3].

16. Ordenando cositas

Haz un programa en Python que ordene una lista de tuplas. Cada tupla tiene dos elementos. La lista debe quedar ordenada de mayor a menor por el segundo elemento de cada tupla.

17. Sudokus

Haz un programa que lea de un fichero un Sudoku, resuelto. Es decir, una tabla de 9 x 9 números enteros del 1 al 9 (todas las casillas están cubiertas). El programa debe decir si el sudoku está resuelto correctamente o no.

18. Matricitas

Haz un programa de Python que lea dos matrices de números enteros y muestre por pantalla el resultado de multiplicarlas.

19. Contando valores

Haz una función en Python que reciba un diccionario por parámetro y devuelva el valor más frecuente de dicho diccionario y las claves en las que se encuentra.

20. Suma de los números de una frase

Haz una función en Python que reciba una cadena de texto por parámetro y que devuelva la suma de todos los números que haya en esa cadena de texto.

Por ejemplo: para la cadena de texto "Tengo 5 gatos con 4 patas cada uno. Uno tiene 12 años." habría que devolver el valor 21.

21. Tecleando

Haz una función que reciba una lista de palabras por parámetro. La función debe devolver otra lista con aquellas palabras que pueden ser escritas en un teclado QWERTY utilizando solo una fila del teclado.

La primera fila de un teclado QWERTY tiene las letras "qwertyuiop", la segunda tiene "asdfghjklñ", y la tercera tiene "zxcvbnm".

22. Aplanando listas

Haz una función en Python que reciba una lista cuyos elementos pueden ser números o listas de números. A su vez, cada sublista puede contener también listas de números.

Tu función debe “aplanar” la lista recibida para que sea una lista simple de números.

Ejemplo: de `[1, [2, 3], [4, [5, 6, [7]]]]` a `[1, 2, 3, 4, 5, 6, 7]`.

23. Ordenación rápida

Haz una función en Python que reciba una lista de números por parámetro y la ordene. Implementa para ello el algoritmo quicksort.

24. Buenos pares

Haz una función en Python que reciba una lista de números por parámetro y devuelva el número de buenos pares que contiene.

Un buen par de números está definido por las siguientes condiciones:

- `numeros[i] == numeros[j]`
- `i < j`

25. Números romanos

Haz un programa en Python que convierta de números romanos a enteros y de enteros a números romanos.

26. A pedradas

Tenemos una lista de números enteros que representan los pesos de N piedras.

Ahora vamos a jugar a un juego. Se trata de tomar las dos piedras más pesadas y hacerlas chocar entre sí.

Si los pesos de las piedras son iguales, ambas quedan destruidas. Si no, la más ligera queda destruida y la mayor pasa a tener el peso que tenía restándole el de la más ligera.

Las piedras destruidas se eliminan de la lista.

Y así, se sigue jugando hasta que solo queda una piedra. O ninguna, que puede suceder.

Haz una función en Python que, para una lista con los pesos de las piedras, devuelva el peso final de la piedra ganadora. O cero si no queda ninguna.

27. Bombones

Tienes una lista de números enteros que representa el precio de N bombones. También tienes una cantidad entera D de dinero.

Tienes que comprar exactamente dos bombones de forma que te sobre la mayor cantidad (no negativa) de dinero.

Haz una función que, para una lista de precios de bombones y una cantidad de dinero D, devuelva la cantidad que sobra después de hacer la compra. Si no hay manera de comprar dos bombones, devuelve D.

28. Detectando transacciones inválidas

Una transacción es probablemente inválida si excede de 1000\$ o si ocurre dentro de un intervalo de 60 minutos con otra transacción con el mismo nombre pero en diferente ciudad.

Tenemos una lista de cadenas de texto que consisten en valores separados por comas y que representan, en este orden, el nombre de la transacción, el tiempo (en minutos desde la primera transacción), la cantidad en \$ y la ciudad. Por ejemplo:

`["Juan,0,500,Madrid", "Rubén,30,750,Barcelona"]`.

Haz una función que para una lista de transacciones devuelva aquellas que son probablemente inválidas.

29. Eliminando dígitos

Dada una cadena de texto que representa un número entero N no negativo y un número entero K, haz una función que devuelva el número entero más pequeño resultante de eliminar K dígitos de N.

30. Caramelos

Tienes una lista de N números enteros. Cada número representa la calificación de un niño.

Les vas a repartir caramelos de forma que cada niño tenga al menos un caramelo. Además, un niño debe tener más caramelos que sus vecinos (los que están junto a él en la lista) si tiene una calificación más alta que ellos.

Haz una función que, para la lista de calificaciones de niños, devuelva el número mínimo de caramelos necesarios que hay que repartir para que se cumplan las condiciones.

Comentarios finales

¿Lo has logrado? Espero que sí.

Si has realizado todos estos ejercicios has avanzado un camino enorme desde que te descargaste este fichero por primera vez. ¡Enhorabuena!

¿Cómo te sientes? Seguro que maravillosamente bien.

Con estos ejercicios has practicado tres pilares fundamentales de la programación en Python:

- 1) La bases de la programación
- 2) Las características del lenguaje
- 3) La resolución de problemas

Mi sugerencia ahora es que repitas (sí, repetir es mejor de lo que crees) estos ejercicios pero prestando especial atención al punto 2, las características de Python.

Encuentra la manera más “pitónica” de resolver estos ejercicios.

¡Feliz programación!

Actualización e información IMPORTANTE

Debido a las consultas y peticiones recibidas sobre estos ejercicios, he tomado la decisión de incluir mis propias soluciones a estos ejercicios como un minicurso en la Academia de Código Pitón.

El precio es ridículo, simbólico, para facilitar el acceso y que no suponga un problema para nadie.

Tienes más información acerca de esto en el siguiente enlace, donde además puedes adquirir las soluciones.

[30 Ejercicios de Python Resueltos](#)