

## WHILE (Python)

El ciclo while en Python es una estructura de control de flujo que permite repetir un bloque de código mientras se cumpla una condición. En general, la sintaxis del ciclo while es la siguiente:

while condicion:

    # bloque de código que se ejecuta mientras se cumple la condición

En esta estructura, la condición es una expresión booleana que se evalúa en cada iteración del ciclo. Si la condición es verdadera, se ejecuta el bloque de código que está indentado debajo de la línea del while. Después de ejecutar el bloque de código, se vuelve a evaluar la condición, y el ciclo se repite si la condición sigue siendo verdadera. Si en algún momento la condición se evalúa como falsa, el ciclo termina y la ejecución del programa continúa con la siguiente línea después del bloque del while.

Es importante tener en cuenta que si la condición nunca se cumple, el ciclo while se ejecutará de forma indefinida, lo que puede llevar a problemas de rendimiento o a errores en el programa. Por lo tanto, es fundamental asegurarse de que la condición sea modificada dentro del bloque de código del ciclo while para evitar un bucle infinito.

Además, en Python, el ciclo while se puede combinar con otras estructuras de control de flujo, como los condicionales if-else y los ciclos for, para crear programas más complejos y versátiles.

## GUIA EJERCICIOS BASICOS WHILE

1. Dado un número entero  $n$ , imprimir todos los números desde  $n$  hasta 1 en orden descendente.
2. Dado un número entero  $n$ , imprimir todos los números desde 1 hasta  $n$  en orden ascendente.
3. Dada una cadena de texto, imprimir cada uno de los caracteres en la cadena.
4. Dada una lista de números, imprimir la suma de todos los elementos de la lista.
5. Dado un número entero  $n$ , imprimir si el número es primo o no.
6. Dado un número entero  $n$ , imprimir la suma de todos los números pares menores o iguales a  $n$ .
7. Dada una lista de números, imprimir todos los números que son mayores que el promedio de la lista.
8. Dada una cadena de texto, imprimir la cantidad de vocales en la cadena.
9. Dado un número entero  $n$ , imprimir todos los números impares menores o iguales a  $n$ .
10. Dada una lista de números, imprimir la cantidad de números pares en la lista.

11. Dada una lista de palabras, imprimir todas las palabras que tengan una longitud mayor a 5 caracteres.
12. Dado un número entero  $n$ , imprimir la suma de todos los números impares menores o iguales a  $n$ .
13. Dada una lista de números, imprimir la cantidad de números negativos en la lista.
14. Dada una cadena de texto, imprimir la cantidad de veces que aparece una letra específica en la cadena.
15. Dado un número entero  $n$ , imprimir todos los números primos menores o iguales a  $n$ .
16. Dada una lista de números, imprimir todos los números que son múltiplos de 3.
17. Dada una cadena de texto, imprimir la cadena al revés.
18. Dado un número entero  $n$ , imprimir la suma de todos los números que son múltiplos de 5 menores o iguales a  $n$ .
19. Dada una lista de números, imprimir todos los números que son mayores que el número anterior en la lista.
20. Dado un número entero  $n$ , imprimir todos los números perfectos menores o iguales a  $n$ . Los números perfectos son aquellos números enteros positivos que son iguales a la suma de sus divisores propios (excluyendo al propio

número). En otras palabras, si sumamos todos los divisores propios de un número (excluyendo el número en sí mismo) y el resultado es igual al número, entonces ese número se considera un número perfecto.

Por ejemplo, el primer número perfecto es 6, ya que sus divisores propios son 1, 2 y 3, y  $1 + 2 + 3 = 6$ . Otros ejemplos de números perfectos son 28, 496 y 8128. Actualmente se conocen más de 50 números perfectos, y se cree que existen infinitos números perfectos, aunque no se ha podido demostrar matemáticamente esta afirmación.