

Uso de bucles while y for en Python

- Creado por: Carly Díaz Gutiérrez
- Fecha de realización: 19/03/24

Introducción:

- **Descripción del ejercicio:** En el corazón del aprendizaje de Python, donde la lógica y la creatividad se unen, surge un desafío: escribir dos programas para imprimir los números pares del 1 al 20. Un programa utilizará la fuerza del bucle while, mientras que el otro se moverá con la fluidez del bucle for.
- Este proyecto presenta un algoritmo simple y eficiente para imprimir números pares del 1 al 20 utilizando un bucle for y otro utilizando un bucle while. El algoritmo se describe en detalle, incluyendo su pseudocódigo, diagrama de flujo y explicación en lenguaje natural.
- El proyecto también incluye una implementación del algoritmo en **Python**. Se ha probado la implementación y se ha verificado que funciona correctamente.

Desarrollo:

Programa usando bucle **while**:

Pseudocódigo:

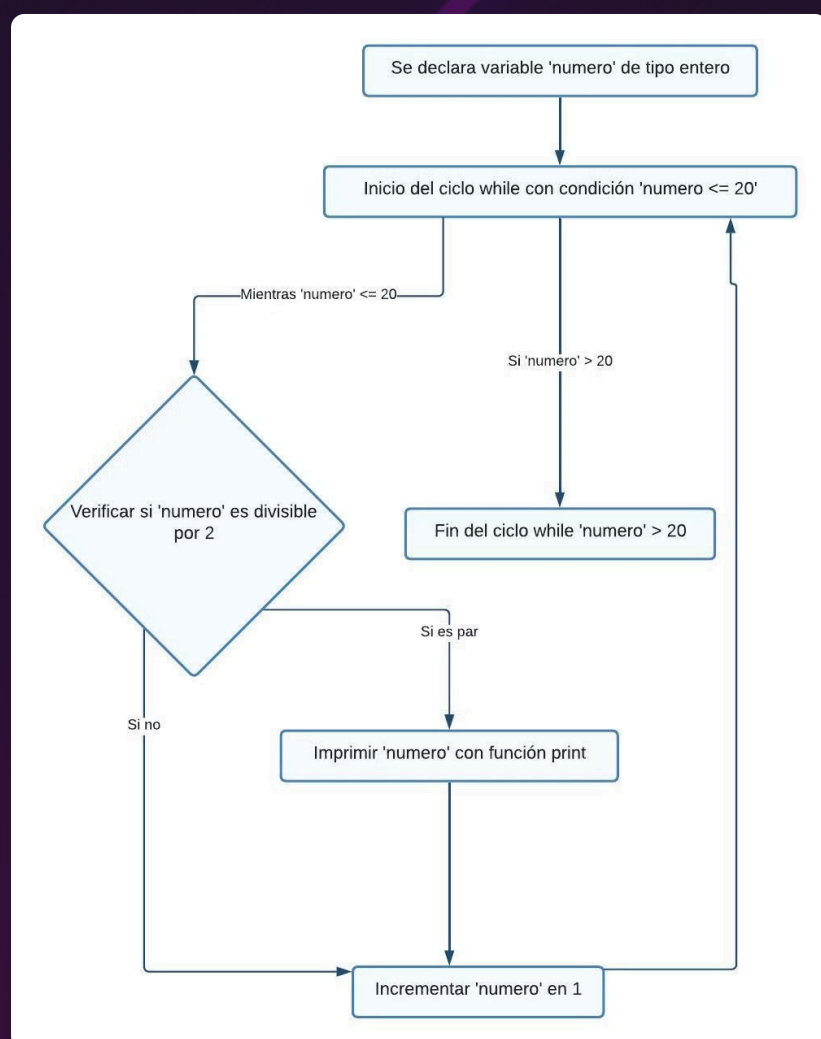
Si el número es par, imprimirlo `if numero % 2 == 0: print(numero)` # Incrementar el número en 1
`numero += 1`

- El bucle **while** se ejecuta mientras el número actual sea menor o igual que 20. En cada iteración, se verifica si el número actual es par. Si lo es, se imprime. Luego, se incrementa el número en 1.

Diagrama de flujo:

Explicación:

- Se declara una variable `numero` de tipo entero.
- Se inicia el ciclo `while` con la condición `numero <= 20`.
- Dentro del ciclo, se verifica si `numero` es divisible por 2 usando el operador `%`.
- Si `numero` es par, se imprime usando la función `print`.
- Se incrementa `numero` en 1 al final de cada iteración.
- El ciclo `while` termina cuando `numero` es mayor que 20.



Código Python:

```
# Programa para imprimir números pares del 1 al 20 usando bucle while

numero = 1 # Inicializar variable con el primer número

while numero <= 20:
    # Verificar si el número es par (residuo de la división por 2 es cero)
    if numero % 2 == 0:
        print(numero)
    numero += 1 # Incrementar el número en 1 para la siguiente iteración
```


Programa usando bucle **for**:

Pseudocódigo:

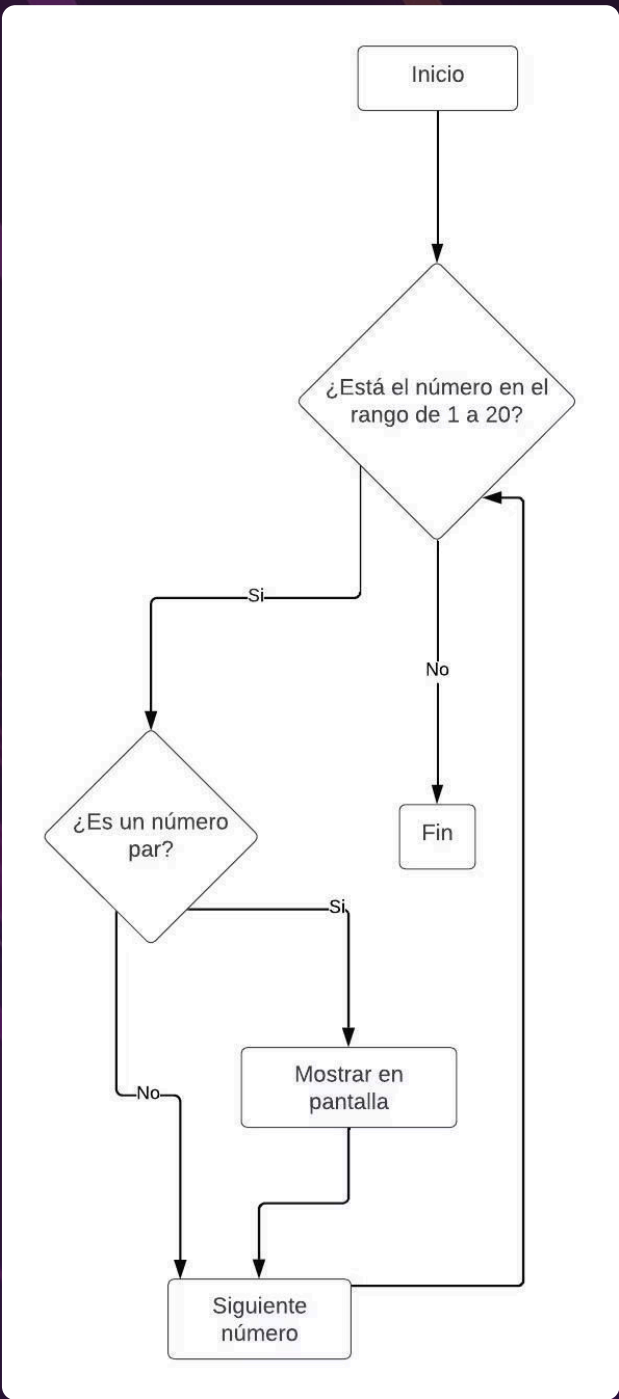
para numero en rango(1, 20): # Si el número es par, imprimirlo if numero % 2 == 0: print(numero)

- El bucle **for** itera desde 1 hasta 20. En cada iteración, se verifica si el número actual es par. Si lo es, se imprime.

Diagrama de flujo:

Explicación:

- Se declara una variable numero de tipo entero.
- Se inicia el ciclo while con la condición numero <= 20.
- Dentro del ciclo, se verifica si numero es divisible por 2 usando el operador %.
- Si numero es par, se imprime usando la función print.
- Se incrementa numero en 1 al final de cada iteración.
- El ciclo while termina cuando numero es mayor que 20.



Código Python:

```
# Programa para imprimir números pares del 1 al 20 usando bucle while

numero = 1 # Inicializar variable con el primer número

while numero <= 20:
    # Verificar si el número es par (residuo de la división por 2 es cero)
    if numero % 2 == 0:
        print(numero)
    numero += 1 # Incrementar el número en 1 para la siguiente iteración
```


Conclusiones:

En este trabajo, se han implementado dos programas en Python para imprimir los números pares del 1 al 20: uno utilizando un bucle while y otro utilizando un bucle for.

Ambos programas son funcionales y cumplen con el objetivo de imprimir los números pares del 1 al 20. Sin embargo, cada uno tiene sus propias ventajas y desventajas:

Bucle while:

- **Ventajas:**
 - Es más flexible y adaptable a diferentes situaciones.
 - Permite realizar comprobaciones adicionales antes de imprimir un número.
- **Desventajas:**
 - Puede ser menos eficiente que el bucle for en este caso particular.
 - Requiere más líneas de código para implementarse.

Bucle for:

- **Ventajas:**
 - Es más conciso y eficiente para este tipo de tarea.
 - Requiere menos líneas de código para implementarse.
- **Desventajas:**
 - Es menos flexible que el bucle while.
 - No permite realizar comprobaciones adicionales antes de imprimir un número.

En este caso, se considera que el bucle for es la mejor opción para imprimir números pares del 1 al 20. Es más conciso, eficiente y fácil de entender.

Además de los dos programas, se han incluido diagramas de flujo y pseudocódigo para facilitar la comprensión del trabajo realizado. También se han utilizado comentarios para explicar la lógica del código y las buenas prácticas de programación.

En general, este trabajo ha demostrado la capacidad de utilizar bucles while y for en Python para resolver un problema específico. Se han aplicado las buenas prácticas de programación y se ha documentado el código de forma clara y concisa.

En resumen:

- El algoritmo es simple, eficiente y fácil de entender.
- Se puede modificar para adaptarlo a diferentes necesidades.
- Es una herramienta útil para aprender sobre bucles for y programación en general.

Recomendaciones:


- Se recomienda practicar la implementación del algoritmo en diferentes lenguajes de programación.
- Se puede explorar la creación de algoritmos más complejos que utilicen bucles for para diferentes tareas.

Recursos adicionales:

- Documentación oficial de Python: <https://docs.python.org/3/>
- Tutoriales de Python: <https://www.learnpython.org/>
- El bucle for en Python: estructura y ejemplos: <https://j2logo.com/bucle-for-en-python/>

Bucle for en Python:



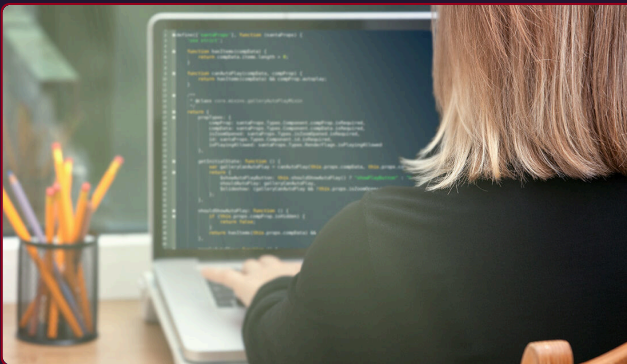
 freeCodeCamp.org


[🔗](#)

Bucle for en Python: Ejemplo de for i en rango

Artículo original escrito por: Jeremy L Thompson

[https://www.freecodecamp.org/news/author/jeremyt/] Artículo...

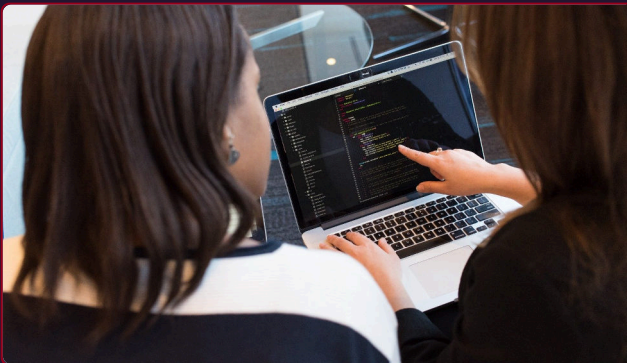



 IONOS

[🔗](#)

Bucle for en Python: comprender y utilizar de forma ade...

A igual que en otros lenguajes, el bucle for forma parte del repertorio básico de programación en Python. Pero, esta...

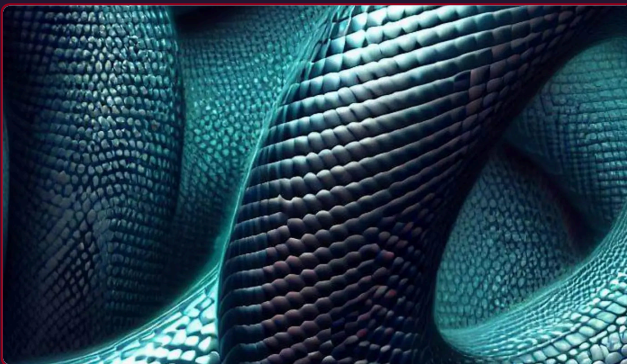



 blog.hubspot.es

[🔗](#)

Bucle for en Python: cómo se utiliza, ventajas y ejemplos

Analiza qué es el ciclo for en Python, su indentación, objetos iterables, ventajas y más.




 El Blog Del Programador

[🔗](#)

Cómo usar el bucle for en Python: Ejemplo con range()

Aprende a iterar en una secuencia utilizando el bucle for en Python y combinarlo con otras estructuras de control de flujo.


 gyata

[🔗](#)

Bucle for en Python: Una guía completa | Programación en Python

Aprenda a utilizar el bucle for en Python. Exploramos su sintaxis, usos, variantes y ejemplos para mejorar tus habilidades de programación.

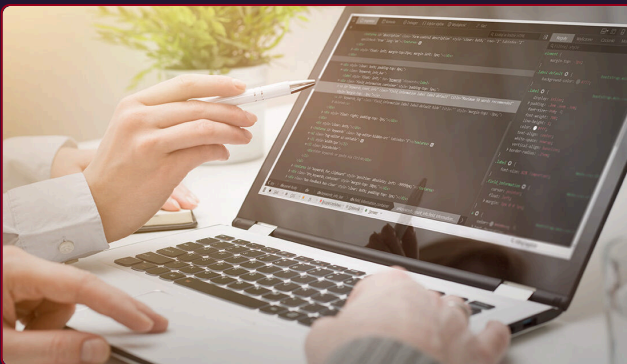
Bucle while en Python:


 www.mclibre.org

[🔗](#)

Bucle while. Python. Bartolomé Sintés Marco. www.mclibre.org

Un bucle while permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor True).




 IONOS

[🔗](#)

Bucle while de Python

El bucle while de Python hace que se ejecute un bloque de código repetidamente mientras una condición sea verdadera.




 KeepCoding Bootcamps

[🔗](#)

Uso del bucle while en Python | KeepCoding Bootcamps

Un bucle while en Python es ideal cuando no sabes cuántas veces necesitas repetir una acción, pero sabes bajo qué...



 blog.hubspot.es

[🔗](#)

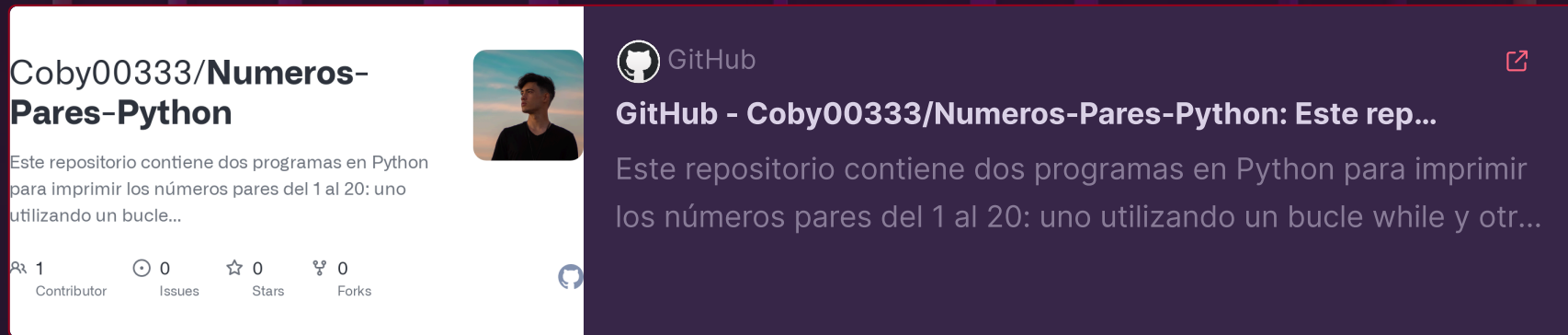
Bucle while en Python: qué es y cómo se usa (ejemplo)

Conoce qué es y cómo hacer la implementación del ciclo while en Python.

Dominio público

Este documento, al igual que el código fuente del proyecto, se encuentra en dominio público. Esto significa que cualquiera puede utilizarlo, modificarlo y distribuirlo libremente, sin necesidad de pedir permiso o dar crédito.

El proyecto se ha subido a GitHub para facilitar su acceso y colaboración. Puedes encontrar el repositorio en el siguiente enlace:



The image shows a screenshot of a GitHub repository page. On the left, the repository name 'Coby00333/Numeros-Pares-Python' is displayed in bold. Below it, a description states: 'Este repositorio contiene dos programas en Python para imprimir los números pares del 1 al 20: uno utilizando un bucle...'. At the bottom of this section, there are statistics: 1 Contributor, 0 Issues, 0 Stars, and 0 Forks. To the right of the repository name is a profile picture of a person. On the right side of the screenshot, there is a dark blue banner with the GitHub logo and the text 'GitHub - Coby00333/Numeros-Pares-Python: Este rep...'. Below this banner, the same description is repeated: 'Este repositorio contiene dos programas en Python para imprimir los números pares del 1 al 20: uno utilizando un bucle while y otr...'. A small red square icon is visible in the top right corner of the banner.

¡Esperamos que este proyecto te sea útil!