**existing library function of gcd uses euclidean algorithm?**

Yes, the existing Python library function math.gcd uses the **Euclidean algorithm** to compute the Greatest Common Divisor (GCD) of two integers. The Euclidean algorithm efficiently finds the GCD by repeatedly applying the operation:

until , at which point the result is .[[1]](#fn1)[[2]](#fn2)

**How math.gcd works internally:**

* It takes two non-negative integers.
* Uses recursive or iterative Euclidean algorithm steps.
* Returns the largest positive integer that divides both numbers without leaving a remainder.
* Is efficient with time complexity approximately .

**Example Python code showing Euclidean algorithm (like in math.gcd):**

def gcd(a, b):  
 return a if b == 0 else gcd(b, a % b)  
  
# Example usage:  
import math  
a, b = 60, 48  
print(gcd(a, b)) # 12  
print(math.gcd(a, b)) # 12

The built-in math.gcd is optimized and implemented in C internally, but conceptually it follows this Euclidean approach.[[2]](#fn2)[[3]](#fn3)[[1]](#fn1)

⁂

1. <https://www.geeksforgeeks.org/python/gcd-in-python/>

1. <https://www.w3schools.com/python/ref_math_gcd.asp>

1. <https://www.geeksforgeeks.org/dsa/program-to-find-gcd-or-hcf-of-two-numbers/>

1. <https://stackoverflow.com/questions/11175131/code-for-greatest-common-divisor-in-python>

1. <https://www.ccbp.in/blog/articles/gcd-of-two-numbers-in-python>

1. <https://prepinsta.com/python-program/find-gcd-of-two-numbers/>

1. <https://www.youtube.com/watch?v=cahuG1cEQdY>

1. <https://www.programiz.com/python-programming/examples/hcf>