

ChatGPT

88 GPTs erkunden

Python GCD Calculation

GCD Calculation in Python

Finding GCD in Python

Bedürfnisse in der Karriere

Al in Education Research Paper

Hectic Meeting with Youngsters

Masterarbeitsthema Diskussion Ter

Stay in Bari!

StratDiag & Tech: Structuring

New Apollo Software Packages

Dictionary Encoding in Databases

Committees Relevant in Ambition.

Update Dictionary and Vectors

SQL CRUD with Dictionary-Encodir

Neuer Chat

Please consider this programming problem: Given two positive integers A and B, find GCD of A and B. Provide an example solution for the problem in the Python language.

Certainly! To solve the problem of finding the Greatest Common Divisor (GCD) of two positive integers A and B, we can use the Euclidean algorithm. This algorithm is an efficient way to compute the GCD, based on the principle that the GCD of two numbers also divides their difference.

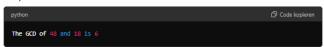
Here's the step-by-step Python code to solve the problem:

```
def gcd(a, b):
      a, b = b, a % b
# Example usage:
B = 18
result = gcd(A, B)
```

Explanation:

- 1. Euclidean Algorithm:
 - $\bullet \quad \text{Start with two numbers A and B.}$
 - Replace A with B, and B with A modulo B.
 - \bullet Repeat this process until B becomes 0. The GCD is the last non-zero remainder.
- 2. Example:
 - Let's find the GCD of 48 and 18.
 - First step: A = 48, B = 18.
 - Calculate $48 \mod 18 = 12$.
 - $\bullet \quad \text{Replace A with 18 and B with 12.} \\$
 - Calculate $18 \mod 12 = 6$.
 - Replace A with 12 and B with 6.
 - Calculate $12 \mod 6 = 0$.
 - \bullet $\;$ Since B is now 0, the GCD is the current value of $A\!,$ which is 6.

Output:



This code works efficiently even for large values of A and B. The Euclidean algorithm has a time complexity of $O(\log(\min(A,B)))$, making it very efficient.

Team-Arbeitsbereich hinzufügen

Sende eine Nachricht an ChatGPT

ChatGPT kann Fehler machen. Überprüfe wichtige Informationen.