

# LU04.L10 - Größter gemeinsamer Teiler mit reduce

```
from functools import reduce

# Der Algorithmus von Euklid ist etwas zu lange um ihn nur als Lambda-
# Funktion zu schreiben.
# Deshalb schreiben wir eine Funktion, die den Algorithmus implementiert.
def euklid(a, b):
    """
    Berechnet den größten gemeinsamen Teiler von a und b.

    Args:
        - a (int): Eine Zahl.
        - b (int): Eine Zahl.

    Returns:
        - int: Der größte gemeinsame Teiler von a und b.
    """
    if b == 0:
        return a
    return euklid(b, a % b)

def gcd(numbers):
    """
    Berechnet den größten gemeinsamen Teiler einer Liste von Zahlen.
    Benutzt dazu die Funktion euklid(a, b) mit reduce().

    Args:
        - numbers (list): Eine Liste von Zahlen.

    Returns:
        - int: Der größte gemeinsame Teiler der Liste.
    """
    return reduce(euklid, numbers)

if __name__ == '__main__':
    numbers = [12, 15, 21]
    result = gcd(numbers)
    print(result) # Sollte 3 ausgeben
```



© Kevin Maurizi

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**



Permanent link:  
<https://wiki.bzz.ch/modul/m323/learningunits/lu04/loesungen/reduce2>

Last update: **2024/03/28 14:07**