

Euklidischer Algorithmus.

```
def euclidean_algorithm(a,b):  
    """Euklidischer Algorithmus zur Bestimmung des größten  
    gemeinsamen Teilers der (ganzen) Zahlen a und b."""  
    # Es sollte  $a > b > 0$  gelten: Wir korrigieren  
    # "fehlerhafte" Eingaben, falls erforderlich.  
    if a == 0:  
        return b  
    if b == 0:  
        return a  
    if a < 0:  
        a = -a  
    if b < 0:  
        b = -b  
    if a < b:  
        # Vertausche a und b, sodass  $a > b$  gilt: Achtung, die Zeilen  
        #  $a = b$   
        #  $b = a$   
        # würden NICHT das gewünschte Ergebnis liefern!!! Möglich wäre:  
        # dummy = a  
        # a = b  
        # b = dummy  
        # Aber in Python geht das einfacher (Zuweisung eines Paars von Werten):  
        a,b = b,a  
    # Die folgende while-Schleife läuft scheinbar "für immer":  
    # Wir springen aber natürlich nach endlich vielen Schritten  
    # heraus, und zwar mit dem return-Befehl.  
    while True:  
        # Division mit Rest:  
        quotient = a // b # NICHT  $a / b$ : Ergebnis i.a. KEINE ganze Zahl!  
        remainder = a % b # remainder = a modulo b,  
        # also  $a = \text{quotient} * b + \text{remainder}$   
        if remainder == 0:  
            # Der letzte auftretende Rest ungleich Null ist b:  
            return b  
        # Implizites else: Im Fall remainder==0 sind wir ja aus der Funktion  
        # (und daher auch aus der while-Schleife) mit dem return-Befehl  
        # "herausgesprungen".  
  
        # Vorbereitung der nächsten Division mit Rest:  
        a = b  
        b = remainder
```