



Consider the following programming problem:  
Given a positive integer  $N$ . Calculate the Fibonacci series till the number  $N$ . If  $N$  is a part of the series, include  $N$  as well.  
Please provide a solution for this problem in the Python language.

Vorschläge anzeigen ▾



Python

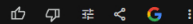
```
def fibonacci_series(N):  
    """Calculates the Fibonacci series up to N.  
  
    Args:  
        N: A positive integer.  
  
    Returns:  
        A list containing the Fibonacci series up to N.  
    """  
  
    fib_series = [0, 1]  
    while fib_series[-1] <= N:  
        next_fib = fib_series[-1] + fib_series[-2]  
        fib_series.append(next_fib)  
  
    return fib_series  
  
# Example usage:  
N = 20  
result = fibonacci_series(N)  
print(result)
```

Verwende den Code [mit Vorsicht](#).

This Python code effectively solves the problem of calculating the Fibonacci series up to a given positive integer  $N$ . Here's a breakdown of the steps involved:

- Function Definition:** The `fibonacci_series` function takes a positive integer  $N$  as input.
- Initialization:** A list `fib_series` is created to store the Fibonacci numbers. It starts with the first two terms of the series: 0 and 1.
- Iteration:** While the last element in `fib_series` is less than or equal to  $N$ :
  - The next Fibonacci number is calculated by adding the last two elements of `fib_series`.
  - The newly calculated Fibonacci number is appended to `fib_series`.
- Return:** The function returns the `fib_series` list, which contains the Fibonacci numbers up to  $N$ .

The example usage demonstrates how to use the function. By calling `fibonacci_series(20)`, the code calculates the Fibonacci series up to 20 and prints the result.



Prompt hier eingeben

