



Please consider this 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.
Provide an example solution for the problem in the Python language.

Vorschläge anzeigen ▾



Python

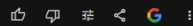
```
def fibonacci_series(n):  
    """Calculates the Fibonacci series up to the given number n.  
  
    Args:  
        n: The maximum number in the series.  
  
    Returns:  
        A list containing the Fibonacci series up to n.  
        """  
  
    fib_series = [0, 1]  
    while fib_series[-1] < 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 calculates the Fibonacci series up to the given number `n`. Here's a breakdown of the steps involved:

1. **Define the function:** The `fibonacci_series(n)` function takes an integer `n` as input and returns a list containing the Fibonacci series up to `n`.
2. **Initialize the series:** The series starts with the first two Fibonacci numbers, 0 and 1.
3. **Calculate subsequent numbers:** The `while` loop iterates as long as the last number in the series is less than `n`. In each iteration, the next Fibonacci number is calculated by adding the last two numbers in the series.
4. **Append the new number:** The calculated number is appended to the end of the series.
5. **Return the series:** Once the loop terminates, the function returns the complete Fibonacci series.

The example usage demonstrates how to call the function with a specific value for `n` and print the resulting series.



Prompt hier eingeben

