Solution Review: Fibonacci Series

This review explains the solution for the 'Fibonacci Series' problem.

We'll cover the following

- Solution
- Explanation

Solution

```
def fib(n):
    # The first and second values will always be fixed
    first = 0
    second = 1

    if n < 1:
        return -1

    if n == 1:
        return first

    if n == 2:
        return second

    count = 3  # Starting from 3 because we already know the first two values
    while count <= n:
        fib_n = first + second
        first = second
        second = fib_n
        count += 1  # Increment count in each iteration
    return fib_n


n = 7
print(fib(n))
```

Explanation

The first thing we need to do is handle all the edge cases. If `n` is less than `1`, we simply have to return `-1`. If `n` is `1` or `2`, that means we have to return the first or second value in the series.

Since we already know these are always `0` and `1`, we can easily check for them at the start.

Now begins our `while` loop. A `for` loop would work just as well. Our iterator is the `count` variable, which starts from `3` because we already handled the first two values in the series.

In every iteration, we use the previous two terms in the sequence, which are `second` and `fib_n`. These become `first` and `second` in the next iteration.

In the end, `fib_n` will hold the final *nth* value in the Fibonacci sequence.