Euler 0004

The Problem:

9009 is a palindromic number since it can be read forwards and backwards the same. It is the product of 91*99 and is the largest palindromic number generated by the product of 2 2-digit numbers.
What is the largest palindromic number that is the product of 2 3-digit numbers?

The approach:

The incredibly naive approach would be to start with 999*999 and iterate through all iterations going down through all the numbers 999->1 and 999->1. We are probably going to find the number pretty near the top, but still running the entire brute force is *only* 998001 iterations... which isn't *too* bad.

As for the checker, we can convert the number into a string and check if it the same forwards and backwards.

The code:

```
largest_palindrome = 0


def palindrome(number):
    return str(number) == str(number)[::-1]


for i in range(1000,1,-1):
    for j in range(1000,1,-1):
        if palindrome(i*j) and i*j > largest_palindrome:
            largest_palindrome = i*j


print(largest_palindrome)
```