

Scientific Programming (Wissenschaftliches Programmieren)

Exercise 2

1. IPython notebooks

- Familiarise yourself with the web-based IPython notebook
- Make some trivial arithmetic operations in the notebook, save it, open it again, etc.
- Find out how you can insert text (raw text, headings, etc.) into your notebook

2. Palindrome

- Create a Python-script / IPython-notebook which checks whether a given word is a palindrome.
- The script should read a phrase as input and print a message whether it is a palindrome.
- It should repeatedly read a phrase and make the check as long as the entered phrase is not empty (the user hits only Enter when entering the phrase).

3. Fibonacci numbers

- Create a Python-script / IPython-notebook which generates Fibonacci numbers.
- The first two elements of the Fibonacci series are 1, otherwise each element is the sum of the previous two elements.
- The script should read the number of desired Fibonacci numbers as input, then calculate and print the Fibonacci numbers (one at a time; the printed numbers should be aligned to the right with a field width of 10 characters).
- Make sure the script also works correctly, when the desired number of Fibonacci terms is only one or two.

4. Prime check

- Create a Python-script / IPython-notebook which checks whether a number is a prime number.
- The script should read a positive integer as input, and then check for the necessary range of numbers whether they are divisor of the input or not. (Try to find the minimal range of numbers to check!)
- It should print a message telling whether the number is a prime or not.
- Make sure, the script also works correctly when the numbers 0 or 1 are entered.

5. Prime factorization*

- Create a Python-script / IPython-notebook which factors an integer into primes.
- The script should read a positive integer from standard input.
- It should print the prime factors of the number (one prime factor per line) and how often that prime factor is contained.