Assignment

- Create a function, which takes any number of numbers and returns their sum

- Create a function, which tests whether the string given via the argument is palindrome or not

- Create a function, which generates a random password
  - it takes optional arguments with default values for
    - length of the password
    - number of special characters
  - useful functions to implement it
    - random.randint(a, b)
    - random.choice(sequence)

Assignment

- Implement simplified version of the **map** function
  - a function **my_map** that takes a list and function and applies the function to all elements in the list and returns a new list
  - try your **my_map** function to map a list of strings to a list of strings where each string is reversed
    - [‘one’, ‘two’, ‘three’] -> [‘eno’, ‘owt’, ‘eerht’]

Assignment

- 1st version: Create fibo_gen(n) generator, which produces Fibonacci numbers till the n-th one

- 2nd version: Create fibonacci(n) generator, which produces unlimited sequence of Fibonacci numbers
  - using it, implement
    - fib(n) function returning n-th Fibonacci number
    - fibo_gen(n) generator

Assignment

- Create your own version of **range()**
  - range is a generator
  - can be called with 1 or 2 or 3 arguments
    - range(limit)
    - range(start, limit)
    - range(start, limit, step)

Assignment

- Write a program, which displays files line by line
  - the list of files is specified as a command line argument
  - after displaying one line, the program waits for user input – the user can:
    - press Enter to display the next line
    - press n + Enter to forget the rest of the current file and start with the next file
    - press q + Enter to terminate the program
    - or anything else + Enter to display the next line