Prof. Dr. Natalie Packham
LV-Nr. 425093

Summer term 2019

Computational Finance and FinTech – Problem Set 1 with solutions

Exercise 1. Download and install Python and Jupyter Notebook (e.g. via Anaconda). Make sure you can run the Jupyter Notebook `00_getting_started.ipynb`.

Solution 1. n/a

Exercise 2. Let `name="John Smith"`. What are (explain!)

- `name[1]`,
- `name[-2]`,
- `name[1:-1]`?

How do you refer to all letters except the first one?

Solution 2.

- `'o'`
- `'t'`
- `'ohn Smit'`

Indices start at 0, so 1 is second character. -1 is last character and -2 is second-to-last. $[1:-1]$ refers to indices 1 through -1 excluding -1.
All except first: `name[1:]`.

Exercise 3. What is the result of `f"{2+2}+{10%3}"`? Explain.
(Hint: `https://realpython.com/python-f-strings/`)

Solution 3. `'4+1'`
Prints the string with $2+2$ and $10\%3$ replaced by their results. % refers to modulo, which is the remainder in integer divison.

Exercise 4. Let `name="john smith"`. What does `name.title()` return? Explain.

Solution 4. `'John Smith'`
Letters are capitalised.

Exercise 5. How can you check if `name` contains `"John"`?

Solution 5. The function call `name.find("John")` returns 0, which refers to the index where the substring starts. If the string is not contained in the variable, then -1 is returned.

Exercise 6. If `x=1`, what is the value of `x` after executing `x+=2`? Explain!

Solution 6. `3`
`x+=2` is short-hand for adding 2 to x and assigning it back to x, i.e., it is equivalent to `x=x+2`.

Exercise 7. What is the result of `10=="10"`? Explain!

Solution 7. `False`
First is an integer, second is a string.

Exercise 8. What is the result of `"bag" > "apple"`? Explain!

Solution 8. `True`
String comparison follows lexiographic order. Bag comes after apple in an encyclopedia.

Exercise 9. Write a function that returns the maximum of two numbers.

Solution 9.

```
def maximum(x,y):
    if x > y:
        return x
    else:
        return y
```

Exercise 10. Write a function that takes one integer parameter and returns all prime numbers between 1 and the given argument (the value passed as the parameter).

Solution 10. An efficient method is the "Sieve of Eratothenes", which iterates over all numbers between 2 and $\sqrt{x}$ crossing out all *multiples* of these numbers in the range $[2, x]$.

```
def primes(x):
    p=list(range(2,x+1))
    for i in range(2,(int(x**0.5)+1)):
        j=2*i
        while(j<=x):
            if(j in p):
                p.remove(j)
            j=j+i
    return p
```