

# Homework 3:

## Object-oriented Programming & Regex

Robert Litschko\*  
Symbolische Programmiersprache

Due: Wednesday November 5, 2025, 12:00 (noon)

In this exercise you will:

- Practice object oriented programming.
- Get experience using the `unittest` framework.

You can monitor your progress by calling (from the `src` direcorey:)  
`python3 -m unittest hw03_regex/test_regular_ex.py`

### Exercise 1: Object-oriented programming II [4 points]

For this exercise we will use the solution of last weeks exercise as a starting point. Please implement your solution in the `hw03_regex/document.py` file.

This part of the homework will be graded using unit tests by running:

```
python3 -m unittest -v test_document.py
```

Implement the following methods:

1. Inheritance [1 points]: Modify the class `PDFDocument` to make it inherit methods and attributes from `TextDocument`.
2. Override the constructor in `PDFDocument` [1 points]: It should accept a `docid` and a `filepath` variable (string) that points to the location of a pdf file on disk<sup>1</sup>. You should first use the `load_pdf()` function provided by us to extract the content of the pdf file<sup>2</sup> and then pass the text and the document id to the parent constructor.

---

\*Credit: Exercises are based on previous iterations from Katerina Kalouli.

<sup>1</sup>For example: `/home/usr/myfile.pdf`

<sup>2</sup>For this to work you need to install PyPDF with `'pip install PyPDF2'`

3. Aggregation [2 point]: Create a class `Author` with the attributes `firstname`, `lastname`, and `age`.<sup>3</sup> Add a method `get_initials()` that returns the initials of the author's first and last names in uppercase (e.g., for "John Doe", it should return "J.D."). Extend the constructor of `PDFDocument` by adding an additional parameter and instance attribute `author`.

## Exercise 2: Regular Expressions with Python [5 points]

1. Implement the function `is_valid_email` that takes a string and checks if it's a valid email<sup>4</sup>. Return **True** if it's a valid email and **False** otherwise. [1 points]
2. Complete the function `find_mentions`. This function should accept a string and find all mentions (words preceded by "@"). It should return a list of these mentions. [1 points]
3. Complete the function `redact_mentions`. This function should find all mentions in a text (you can reuse your regex pattern from `find_mentions`) and replaces them with a word of your choice, for example, "[redacted]". [1 points]
4. Write a function `is_strong_password` to check if a provided password is strong. A strong password is defined as one that is at least 10 characters long, contains both uppercase and lowercase characters, has at least one numeral, and does not contain any spaces or tabs. Return **True** if it's strong and **False** otherwise. [1 points]
5. Implement the function `replace_links` that finds all hyperlinks in a text (URLs starting with "http://", "https://") and replaces them with a word of your choice, for example, "[LINK]". [1 points]

## Using NLTK (Optional)

You are able to solve this homework without any external Python-packages. However, the `nltk` package is a widely used text processing library that implements a range of common operations for you. We will see more on NLTK starting from lecture 4, but provide information on how you can already install it in this homework.

You are welcome to install and explore it on your own for solving the above tasks.<sup>5</sup> If you work on the cip pool computers, `nltk` should already be installed. To use the `word_tokenize` function in `nltk`, you may have to download the resource 'punkt':

1. open the Python interactive shell:  
`python3`

---

<sup>3</sup>Implement only the constructor.

<sup>4</sup>For the purpose of this exercise, a valid email consists of a username, an "@" symbol, and a domain.

<sup>5</sup>We will use `nltk` in future lectures and exercises. It's therefore highly encouraged that you familiarize yourself with the package.

2. then execute the following commands:

```
>>> import nltk
>>> nltk.download('punkt')
```

If you use your own computer:

- **Unix (with Python3):**  
sudo apt-get install python3-pip  
sudo pip3 install -U nltk  
Test the installation:  
python3  
>>>import nltk

If you use a virtual environment:

- **Unix venv (with Python3):**  
sudo apt install python3-venv (on debian/ubuntu)  
cd path/my\_group/src  
python3 -m venv venv  
source venv/bin/activate  
pip3 install -U nltk  
Test the installation: python3  
>>>import nltk
- **Anaconda:**  
conda activate myenv  
conda install -c anaconda nltk (or pip install nltk)  
Test the installation: python  
>>>import nltk
- **Windows:** <http://www.nltk.org/install.html>
- **PyCharm:** View > Tools Windows > Python Packages
- **The handling of external Python-packages is a crucial skill!** If you encounter difficulties, ask fellow students or the tutors.