Assignment 1:
Web Crawling

- This assignment is due on **1st November, 2017 (11:00 am, CET)**
- You can discuss the problems with other groups of this course or browse the Internet to get help. However, copy and paste is cheating.
- There are 8 assignments in total. In each one of them, all tasks sum up to 20 points. You need to achieve at least 70% of the points in 7 assignments and at least 50% in the remaining one in order to participate in the final exam.
- Submission at
  https://www.dcl.hpi.uni-potsdam.de/submit/
  - only pdf files
  - one file per group per assignment (assignment1.pdf)
  - put your names and matriculation numbers on *each* page in the pdf file

Task 1: Group Formation

To get started you need to form groups of size two.

a) Get familiar with the submission system
https://www.dcl.hpi.uni-potsdam.de/submit.

b) When you submit your first assignment be sure to include both names of the two members of your group in the author list. **1 P**

Task 2: Information Retrieval Introduction

a) Explain the terms Web search and information retrieval. How do they differ from each other? **4 P**

b) Explain the relevance notion as defined in information retrieval. What makes a document relevant or not relevant to a particular query and user? **3 P**

Task 3: Web Crawling

a) What is the advantage of using HEAD requests instead of GET requests during crawling? When would a crawler use a GET request instead of a HEAD request? **3 P**

b) What are the obstacles that a crawler faces when attempting to fetch web pages? Give examples for challenges concerning the semantic information retrieved and the efficiency of crawling. **4 P**

Task 4: (Programming) Web Crawling

During this course each group will implement their own search engine in Python. At the end of the course we will evaluate all search engines with respect to quality of search results as well as speed and memory consumption. You will build your search engine on newspaper comments. While we provide a list of online newspapers that offer comment sections, one of your tasks will be to crawl comments. Choose your newspaper here: https://doodle.com/poll/

cd7wuhiyd95pcgtm. The programming assignments will guide your development and implementation process. Don't submit any source code for the assignments; just the output of your program as described in the task.

- Download the Python source files from the course's folder and have a look at the example scraper (`WS 2017-18/Excercises/Assignment 1`).
- We provide a small test dataset containing only 10 newspaper comments to build and test your search engine. This file is called testData.csv and is included in the course's folder.
- Later on in the course, you will use your own crawled comment dataset. Thus, you should finish your crawler's code as soon as possible so that you can focus on the other parts of your search engine. Store the data in a simple csv format with `comment_id, article_url, comment_author, comment_text, timestamp, parent_comment_id`. If available, you might include the number of votes for the comment.
- Implement a python method using the provided scrapy template that crawls the newspaper comments for a given date. The method should return a csv file with the beforementioned structure.

a) Print the csv file for one newspaper article of the 16th October, 2017. Choose an article with at least 5 comments. **5 P**