Table of contents:

* Introduction
  + What is the task?
  + Requirements
* Overview of available solutions
  + Different web scraping libraries
    - Requests
    - Beautiful Soup 4 (BS)
      * Chose this because I am most familiar with it
      * More intuitive for me
      * Only need some simple functions
    - Other available:
      * Ixml
      * Selenium
      * Scrapy
    - Regex
  + Different ways to extract the data
    - Regex vs BS methods (or functions)
    - Tags
      * Regex vs BS method
    - Href for web links
      * Regex vs BS method
* Simplified design
  + User input
    - Start URL
    - Crawl debth
    - User defined regexes
  + Find all URLs
  + Find relevant information
    - Phone numbers
    - Emails
    - Words and their occurence
    - Html comments
  + Display the information found
* Detailed implementation explanation
  + Choosing what to print
    - Set the variable to True to output result
  + The functions
    - Find\_all\_URLs() takes url as input, but the rest takes a string
      * This is because of reusability
      * We assume that we always look for URLs on a website
      * We can extract words, numbers, and e-mails from strings as well
    - Find\_all\_URLs()
    - Find\_all\_phone\_numbers()
    - Find\_all\_emails()
    - Find\_all\_words()
  + Taking user input
    - URL validity
    - Crawl debth
    - User defined regexes
  + Finding all URLs
    - Why only URLs first? (and then words, emails etc)
      * Advantage: Better overview, more intuitive
      * Disadvantage: Somewhat slower execution, depending on internet speed
      * Disadvantage: Have to wait untill whole program is finished
    - Searching for href instead of using tags (tag <a>)
      * BS-method doesn’t always catch every tag
      * If you look for tag, it may not contain a url, and is therefore a waste of time
        + It is therefore better to directly search for the href-link
      * Is it faster?
        + Regex is pretty fast, and you don’t have to first extract the tags, and then the content
    - The three lists
      * URLs\_to\_crawl
        + This is the final list of URLs that we are going to scrape for e-mails, phone numbers, and words
      * Next\_debth\_URLs
        + These are the URLs that we are going to look for URLs in if we are going to look one level deeper
      * Found\_URLs
        + This is the list of new URLs we found during the search in the current debth (it’s temporary)
  + The scrape
    - The BS methods
      * URL\_soup.prettify()
        + Returns the html-code as a string
      * " ".join(URL\_soup.stripped\_strings)
        + What is the « «.join() method?
        + What does stripped\_strings() return?
        + Combined, it returns the html content as a string
      * URL\_soup.get\_text() (not used)
    - Finding phone numbers
      * Find\_all\_phone\_numbers(get\_text()) vs re.findall(«href=tel:»)
        + Find\_all\_phone\_numbers searches a string
        + Re.findall(«href=tel:») searches the html code for hyperlinks
    - « «.join(URL\_soup.stripped\_strings) as argument for find words
  + Displaying the data
    - For-loop vs just printing the list
      * I think the for-loop makes it prettier
        + It may be slightly slower
* Analysis and testing
  + Regex vs BS methods
  + Invalid character
    - «Some characters could not be decoded, and were replaced with REPLACEMENT CHARACTER.»
  + Different phone number formats
    - What is a phone number and what is another random number?
  + Access denied on certain cites
* Evaluation
  + Works pretty well
  + Phone numbers are hard
    - Can look for href=tel: but not all numbers are hyperlinked
    - Can look for xx xx xx xx but there are several problems:
      * Phone numbers are written on different forms
      * Phone numbers are written differently by country
      * Some phone numbers include dots and parentheses
    - The trade-off:
      * Only find numbers we are sure of being phone numbers (hyperlink)
      * Find lots of numbers, but unsure they are numbers (free-text search)
    - Phone numbers with and without country codes
  + E-mails are slightly easier
    - Standardized format, making it easier to search in free-text
  + Debth exponentially increases processing time
    - Max 2 recommended