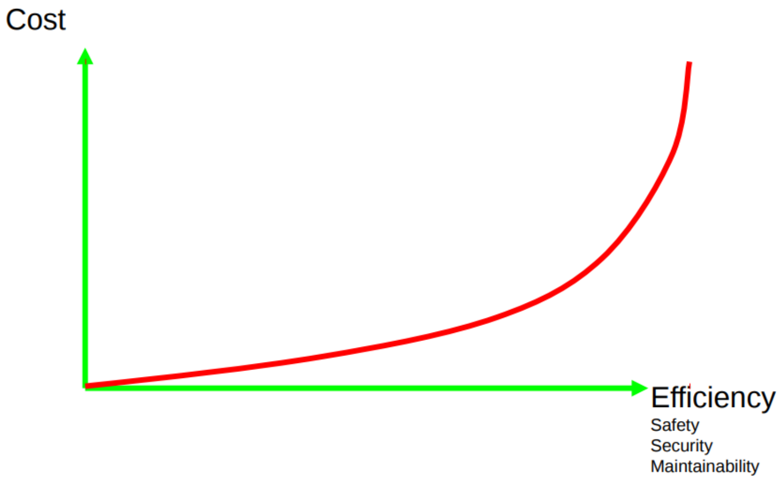
Introduction
Design Quality

- Design quality is an elusive concept
  - Quality depends on specific organizational priorities
- A “good” design may be the most efficient, the cheapest, the most maintainable, the most reliable, etc



https://faculty.washington.edu/ajko/books/cooperative-software-development/quality.html

Software Qualities

- Correctness
  - A program behaves according to its specification.
- Reliability
  - A program behaves the same way over time in the same operating environment.
- Robustness
  - A program can recover from errors or unexpected input.
- Performance
  - A program uses computing resources economically.
- Portability
  - An implementation can run on different platforms without being modified.
- Interoperability
  - A system can seamlessly interact with other systems.
- Security
  - Only authorized individuals can access information in a software system.

Basic Programming Principles

- What are the basic programming principles that every programmer should follow?
  - Keep it simple. Complex code takes longer to design and write
  - DRY: don’t repeat yourself
  - Single Responsibility
    - The single responsibility principle says that every class or module in a program should only concern itself with providing one bit of specific functionality
  - ...

Activity Worksheet!

- Get yourselves into groups of three
- Discuss the worksheet question and write down your answers on each.

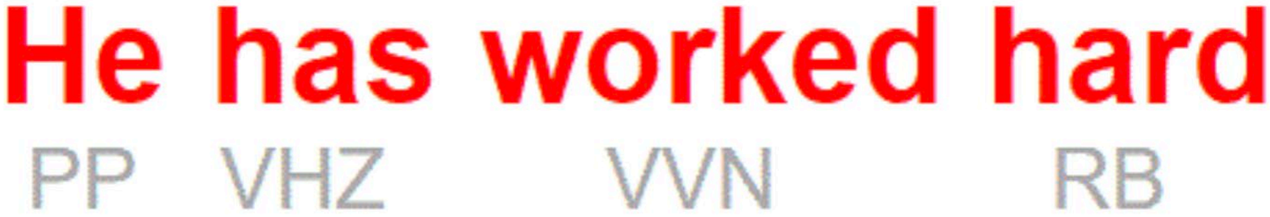
Coderunner - Overview

- About language processing
- For the lab you will have to:
  - Read data from files which contain information about words
  - Process data or sentences to analyse them
  - Calculate statistics about the words in a sentence

- You don’t need to know all the details (e.g. what nouns and verbs are) the lab should walk you though how to do the processing!

Coderunner – Language Processing

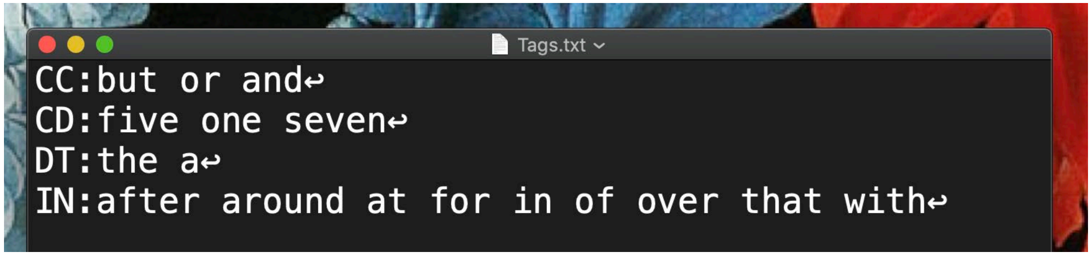
- The words of a sentence can be analysed by their “lexical categories” or “parts of speech” (e.g. nouns, verbs, etc)
- Each of these categories can be given a tag. For example, a personal pronoun can be referenced with the tag “PP”
- By using these tags we can categorise words in a sentence
- For example:



He	has	worked	hard
PP	VHZ	VVN	RB

Coderunner -File and tagging format

- Information for tags will be stored in either files of dictionaries
- If the tag information is in a file, we will need to process it!
    - Files store tags in the following format:
        - <tag>:<words separated by whitespace>\n
        - <tag>:<words separated by whitespace>\n
    - For example:

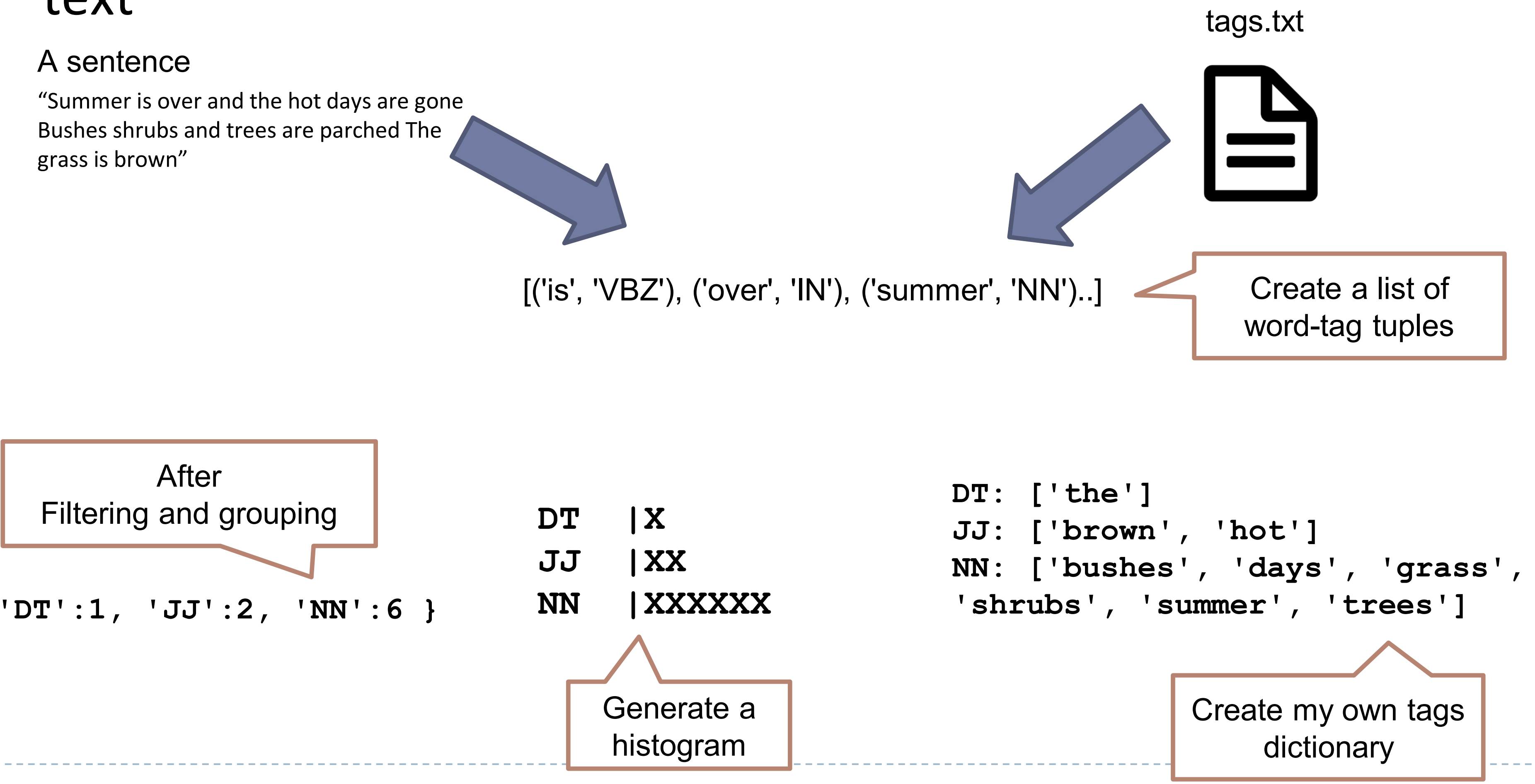


```
CC:but or and↵
CD:five one seven↵
DT:the a↵
IN:after around at for in of over that with↵
```

- We want to turn this file into a dictionary which looks something like:
- {'CC’ : ['and', 'but', 'or’], 'CD’ : ['five', 'one', 'seven’], 'DT’ : ['a', 'the’], 'IN’ : ['after', 'around', 'at', 'for', 'in', 'of', 'over', 'that', 'with’]}
- By creating this dictionary we can begin to analyse sentences

Coderunner – Exercises

- Once we have created our tag dictionary, we can begin analysing text



tags.txt

A sentence

“Summer is over and the hot days are gone Bushes shrubs and trees are parched The grass is brown”

```
[('is', 'VBZ'), ('over', 'IN'), ('summer', 'NN')..]
```

Create a list of word-tag tuples

After Filtering and grouping

```
{'DT':1, 'JJ':2, 'NN':6 }
```

```
DT   |X
JJ   |XX
NN   |XXXXXX
```

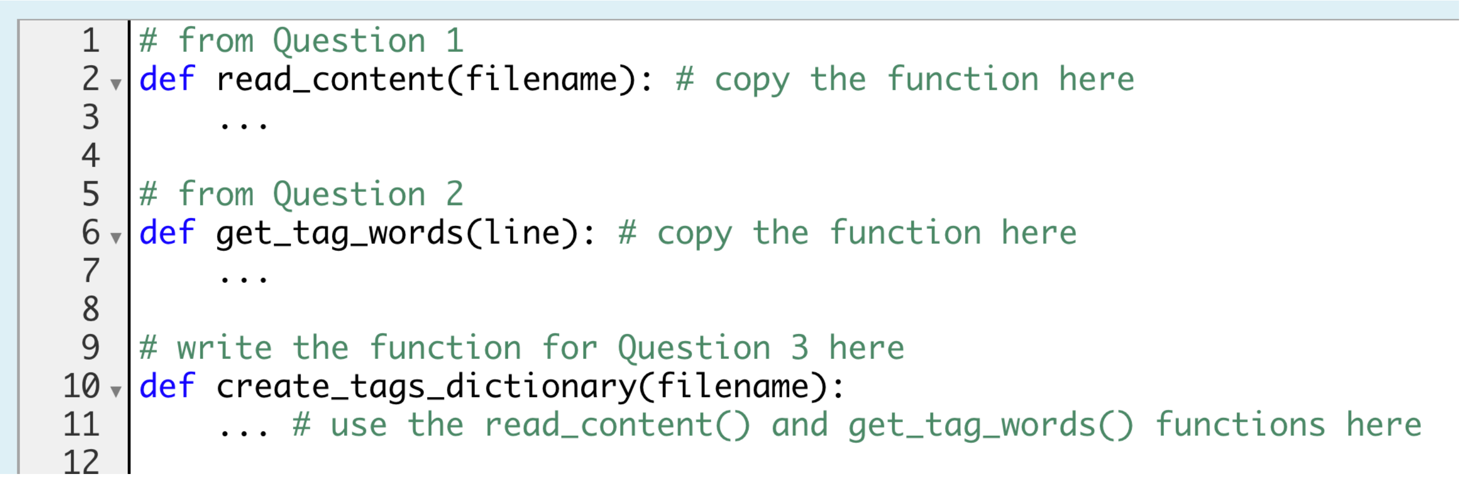
Generate a histogram

```
DT: ['the']
JJ: ['brown', 'hot']
NN: ['bushes', 'days', 'grass',
'shrubs', 'summer', 'trees']
```

Create my own tags dictionary

Coderunner – Tips

- Q3: For this question you have to read the lines from the file, split each line to separate the tags from the words. But! In question 1 and 2 you already wrote functions which do this!
  Your answer should look something like this:



```
# from Question 1
def read_content(filename): # copy the function here
    ...

# from Question 2
def get_tag_words(line): # copy the function here
    ...

# write the function for Question 3 here
def create_tags_dictionary(filename):
    ... # use the read_content() and get_tag_words() functions here

```

Coderunner Tips

- Q6: Just like in Q3, you can copy the functions from Q4 and Q5 into the answer box and they may be able to help.
- Q7: you only need to look at the tags in the tuple list, words can be ignored
- Q9: For this question you will need three for loops inside each other.
- Q10: You will need to select one random index from each of the articles, adjectives, and nouns in that order using the random.randrange(...) function