

This module explains how to process large unstructured documents. Regular expressions help identify patterns in text.

π

Text Editors vs. Word processors

› Text Editor

- Deals with text data only, **no formatting**
- Designed to find information in text, change or replace it


→ for .txt, .log, .bat... files

› Word Processor

- Deals with text data **AND its format**
- Designed to create documents that can be printed in human-readable format.

→ for .odt, .doc, .docx... files

Do not store text data as .doc



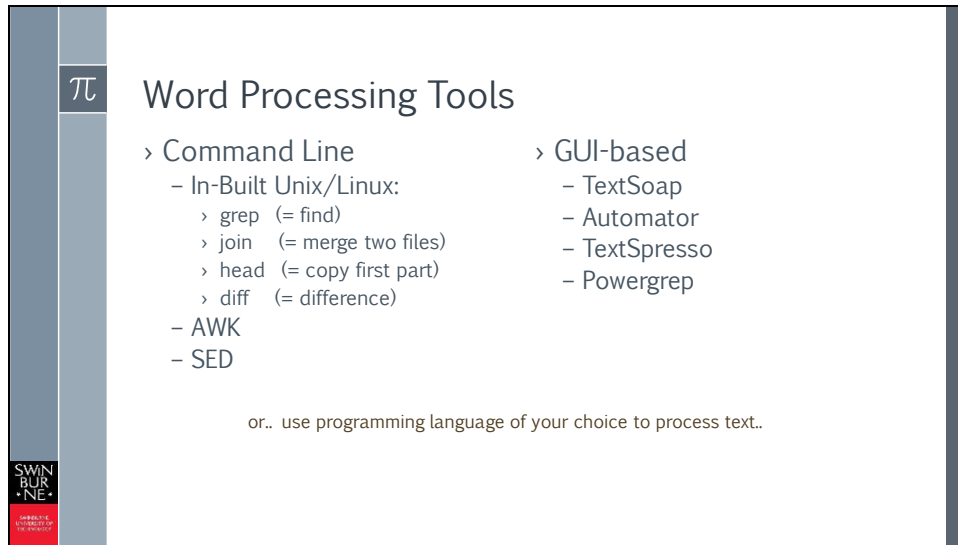
SWINBURNE
UNIVERSITY
TECHNOLOGY

Large documents such as log files are not designed to be humanly readable. When you have to work with them, you need tools that help you find things and possibly copy or replace parts of the file.

For humanly readable text, you often use a word processor. Word processors are designed to add formatting to documents so they become humanly readable.

When you are interested only in extracting data from very large files, word processors are not a good choice. In particular, as Billy says, you should not store your data files in a word processor format. Word processors add a lot of formatting metadata, and it becomes even harder to extract information from these large documents.

Text editors are designed to display text without formatting. They also have many practical tools to find items in the data, replace items or delete items. They also have ways to search using regular expressions, which is a way of finding a pattern – several similar words – instead of one single word. Which is handy when there is slight variation in the way information is expressed.



π

Word Processing Tools

- › Command Line
 - In-Built Unix/Linux:
 - › grep (= find)
 - › join (= merge two files)
 - › head (= copy first part)
 - › diff (= difference)
 - AWK
 - SED
- › GUI-based
 - TextSoap
 - Automator
 - TextSpesso
 - Powergrep

or.. use programming language of your choice to process text..

SWINBURNE
UNIVERSITY OF TECHNOLOGY

From the beginning of times, the Unix operating system, the basis of the Mac and Linux, has offered powerful command-line tools that can search and manipulate large text files. Since MACs and Linux PCs also provide these tools, you may already have them. Even windows offers a command line tool for searching text called findstr.

A large number of GUI-based tools are available on line, many of them free of charge. Most reasonably advanced text editors can handle regular expressions. Some of the tools available can even integrate with word processors.

Any programming language has libraries to access the file system and read text files. It is easy to implement a program that can handle large text files using Java, C++, C#, Perl, Python, Ruby or Javascript.

π Regular Expression

› What is a regular expression?

- A search pattern that can match several literal strings of characters.

British/Australian spelling	American spelling
organisation	organization

organ[is]ation

Regex that matches both

In text, the same concept is often expressed using variations of words with different characters, additional spaces or optional numbers. If you are looking for a particular word, the word might be at the start of a sentence and start with a capital letter. This is why text processors give you the choice of searching with or without regard to upper and lower case characters.

But there are other issues. Sometimes we want to find a word that can be used in a compound (as part of a larger word) or by itself. If we only want to find the occurrences where the word is mentioned on its own, we have to check whether it is followed by a space or a line feed. We also have to check for a space or line feed that precedes the word. If we do this without regular expressions, we have to search for several versions (all combinations of spaces and line feeds before and after the word).

As another example, many words are spelt slightly differently in American and Australian texts. Again, we can search for each version separately, or we can use a regular expression. The word `organ[is]ation` printed in blue has square brackets in the place of the `s` or `z` which contain both characters. This means the characters are optional – we are looking for words that have either `s` or `z` (not both).

π

The Simplest Regular Expression

> ..is a literal string:


star

Find the exact match

The sun is a startling star because it doesn't have a twin.

Most solar systems have a binary star.

Why don't we get less fabricated examples.



SWINBURNE
UNIVERSITY OF TECHNOLOGY

Even when you are trying to find an exact word, you may find similar words you don't want. In this case, you may want to specify that the word has to have a space after it. And before it. Most text editors will let you type a space before and after the search word. But this will not cover line feeds. Or full stops.

π Word Boundaries

- › Beginning of a word: \<
- › End of a word: \>

`\<star\>`

Now we only find the highlighted ones.

The sun is a startling star because it doesn't have a twin.

Most solar systems have a binary star.

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Using the smaller than and greater than characters escaped by backslashes to encase a word make sure we only match the word when it stands on its own. It does not matter whether the word has spaces, full stops or line feeds next to it. But note this version only matches a lower case star. Based on the previous slide, you should know how to write the expression so that it also matches Star when it starts with a capital letter. You may want to pause here and think about it if you haven't already thought of the answer.

When you try this out in your favourite text editor, the regular expression printed here may match upper and lower case letters. If this happens, check whether a box marked 'Match upper and lower case' or similar, is ticked.

π

Options: []

- > [abc]
 - either lower case a or b or c
- > [a-z]
 - any lower case letter of the English alphabet
- > [A-Za-z] (-> not [A-z])
 - any upper or lower case letter of the English alphabet
- > [123]
 - either 1 or 2 or 3
- > [0-9]
 - any digit

ASCII
 A = 65
 Z = 90
 a = 97
 z = 122

Looking for
 actual | or |?
 Use \| or \|

Now that you have found the answer to the question on the previous slide, you already know that square brackets indicate choice. You can list the choices individually inside the brackets. This can give you a very long expression when you want any upper case or lower case letter. Therefore we have the option of noting ranges of values using a dash between the first and last values. The range of characters between a and z is understood because of the binary encoding of the letters. A capital letter A has the number 65 (in binary), a capital Z 90. All other capital letters are between these. This is based on ASCII. If you haven't heard of ASCII, google it, it is worth knowing for computer scientists.


For this discussion, you only need to know that there are a few characters – for example the square brackets and the backslash – between the upper case Z and the lower case a. If you mark the shorthand capital A dash lower case z, these characters are included. You can try this out using your favourite text editor.

Because square brackets have a special meaning, if you actually want to look for square brackets specifically, you have to escape them with backslashes.

π

Examples

- › organi[sz]ation
 - matches organisation and organization
 - does not match organization
- › organi[zs]ation
 - matches organisation and organization
 - does not match organization
- › organi[s]ation
 - matches organisation

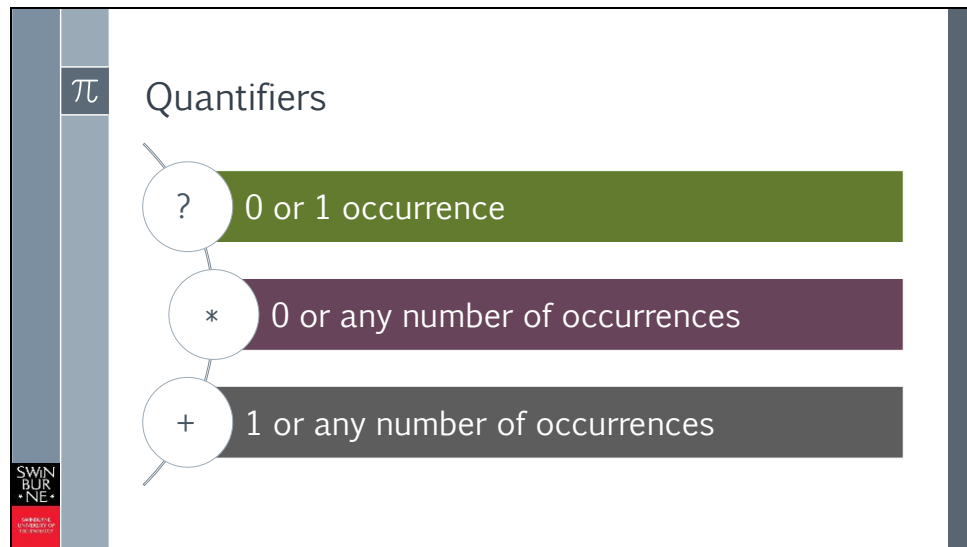


How do we match color and colour?

No choice, so no point using []

SWINBURNE
UNIVERSITY OF TECHNOLOGY

A single occurrence of square brackets matches a single character, one of the options listed in the square brackets. It does not match several characters, or zero characters. If we have s and z in the square brackets, only either s or z is matched at a time. If they both occur in the text, one after the other, this is not a match. This problem comes our way if we want to match the American and the Australian spellings of the word colour or neighbour. Here it is not a choice between two characters; it is either the presence or the absence of a character. So the choices we need to note down are zero or one u. To do this, we have to study quantifiers.



The question mark specifies that a character may or may not be there. It does not allow for several characters.


If we want to specify that a character can be absent, but there can also be any number of such characters, we use the asterisk.

If we want to ensure that there is at least one such character, but there can be more, we use the plus sign.


π

Examples

- › colou?r
 - matches color and colour
 - colo[u]?r works, but [] not necessary
- › UMN ?[0-9]+
 - matches
 - › UMN123,
 - › UMN 14,
 - › UMN 66688,
 - › UMN 8750,
 - › UMN875777, etc.



What if I want exactly 3 numbers?



All we have to do to match both the American and Australian spellings of colour is place a question mark behind the u character. As we know, this means that there can be zero or one u.

Another example is matching number plates. The standard Victorian format is three letters and three digits. If we know what letters we are looking for, we can specify these letters (without allowing any choice). The space between the letters and numbers is optional, so we can add it with a question mark behind the space. If we add the numbers 0-9 in square brackets, we have an option of one number between 0-9. To make sure we allow more than one number, but require at least one, we add a plus behind the square brackets. The problem now is that this matches any number of number characters, so we can have five or a thousand, which is not helpful. Victorian number plates never have more than three numbers. How do we specify this?

π

Matching Exactly n Occurrences


- › UMN `[0-9][0-9][0-9]`

Simple way: Repeat Options

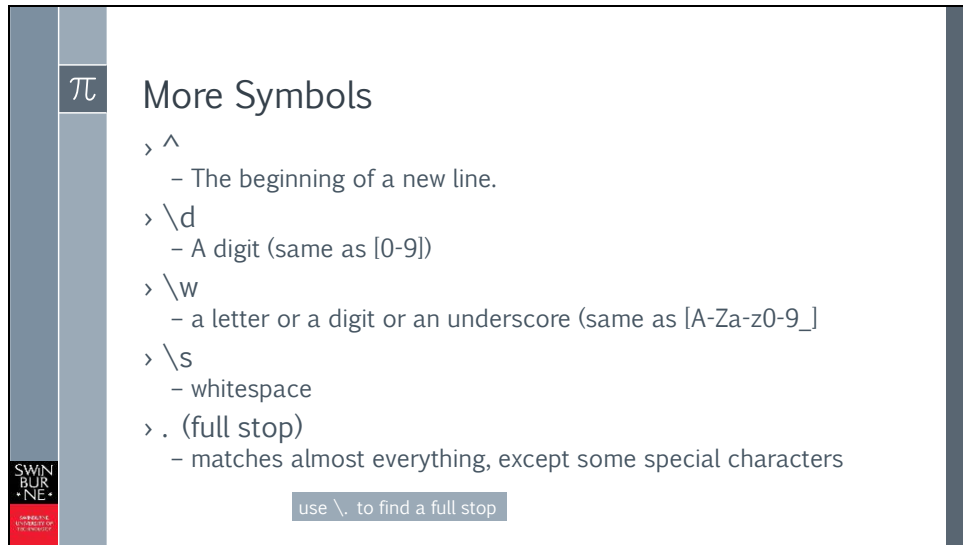
 - matches
 - › UMN123,
 - › UMN 124,
 - › UMN 666,
 - › UMN888, etc.
 - › not UMN 66688, UMN 8750, or UMN875777
- › UMN `[0-9]{3}`

More practical.
- › UMN `[0-9]{2,4}`

Means we allow 2, 3 or 4 digits [0-9]



In the simplest case, we can repeat the options in square brackets as many time as we want them. If you require exactly 25 numbers, this could be a little lengthy. There is a shorthand for this, which is numbers in curly braces. If we mention one number in the curly braces, we want exactly this number of characters. If we put two numbers separated by a comma into the curly braces, this means between a minimum and a maximum. Every number in between is allowed.



π More Symbols

- > ^
 - The beginning of a new line.
- > \d
 - A digit (same as [0-9])
- > \w
 - a letter or a digit or an underscore (same as [A-Za-z0-9_])
- > \s
 - whitespace
- > \. (full stop)
 - matches almost everything, except some special characters

use \. to find a full stop

There are many more symbols in regular expressions, but these are the most important ones.




The escaped letter `\d` can be any digit, so it is shorthand for the 0-9 in square brackets. When we use the capital letter instead, this means everything but the matches for the lower case letter. So `\D` means everything but a number.

The escaped letter `\w` is essentially used for any text – letters, digits and numbers, as well as the underscore. It is useful for finding anything except special characters. Again, `\W` means everything else but letters, numbers and underscores.

The full stop is not very useful because it matches so many characters, but having it as a special character means if you need to find a real point or full stop, you have to escape it with a backslash.

π

Miss Fisher's Murder Mysteries



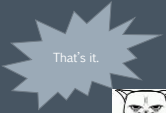




↑	↑	↑	↑	↑
B S A	C O 0	7 1	0 1 2	0 1 2
			9	9

Miss Fisher's Murder Mysteries is a Melbourne-based television series about a private detective, Miss Fisher, and her policeman friend Inspector Robinson, who are chasing murderers in the 1920s.

Right now there has been another murder at Miss Fisher's aunt's house. A guest of the house, a young archaeologist, is found stabbed in his room. His fiancé makes the grizzly discovery and observes a car speeding away in the distance. She caught only a very fleeting glimpse of the number plate. Miss Fisher interviews her and hears that the first letter could have been a B, an S or an A. The second character could have been a C, an O or a zero. The third character was definitely a 7. Or perhaps a 1. There were two more numbers, but it all went too quickly, and the fiancé can't remember. Miss Fisher asks Inspector Robinson to see which ones of these combinations actually exist as registered vehicles. The Inspector quickly calculates that there are 1800 possible combinations and checking all combinations could take all day. Write a regular expression that would make this task really easy.

Summary



- › Unstructured data is here to stay.
- › Unstructured data is often voluminous – there is a lot of it.
- › Searching and manipulating unstructured data is a special challenge.
- › Many tools exist to make this easier.
- › Word processors should only be used for text that is written for human reading. Storing data in such formats adds unnecessary metadata.
- › Finding strings in large data sets is easier when you know regular expressions.

Here are the most important points discussed in this module. You may want to stop the recording to have a read through them. When you are ready, start the quiz about this module.