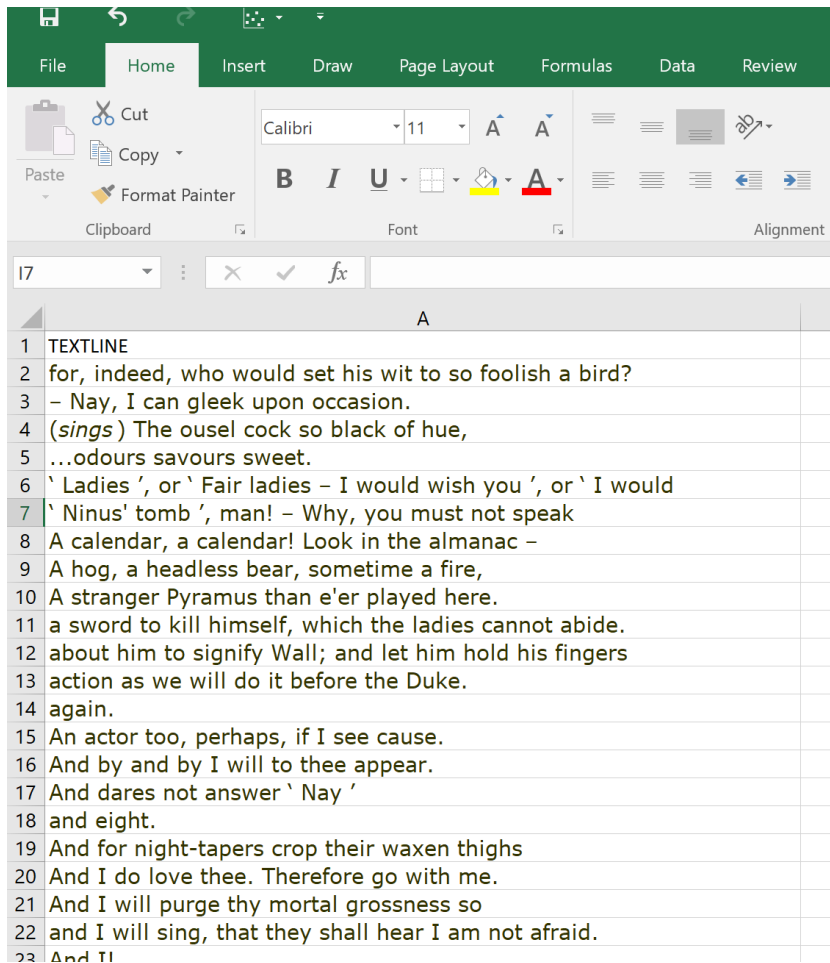


KBL Script for a simple Text Analyser (Word Counts)

This is an example of a working script that analyses text. It isolates individual words and then performs word counts – with a pie chart. You can copy this into your KBL Studio Plus (KSP) and will run – providing you have created a spreadsheet with text to analyse.

In the example the spreadsheet named MSNDAct1.xlsx (Midsummer Night’s Dream Act 1) has the following structure which you can easily replicate.



The KBL Script – You can copy and paste this into a new script:

```
REM -----  
REM KBL Text String manipulation - counting words in text - Analyse Shakespeare  
REM -----  
REM Author PD Date 26/10/2017      Skill Level: Intermediate to Advanced  
REM This example illustrates how to work with text strings in KBL, here we want to look a word usage in a passage of text  
REM First we copy and paste text (e.g. from the web) into a spreadsheet, it will automatically distribute across rows.  
REM We import the spreadsheet into KBL and eliminate blank rows, then remove punctuation marks and create a single string  
REM We locate words by tracking left to right looking for spaces as the markers between words. We count the characters between.  
REM As each word is identified we chop it off the start of the remaining text and put it into a column in MAIN - until we run out of text  
REM The example here is used to analyse Shakespeare's Midsummer Night's dream  
REM If we wanted to we could look for defined words and phrases, count and display them ....a search engine if you will!
```

SECTION Prepare-String

```
// -----
```

```
import /XLSX msndAct1 worksheet=Sheet1  
// Single column spreadsheet created by pasting text lines from the play into first column (Col A)  
// Also removed stage instructions
```

```
// Remove any blank lines read from the spreadsheet  
Mark c.1 = BLANK  
remove /CLEAN
```

```
// Glue the rows together into a single parameter string  
Par intext = " "
```

```
LOOP L1 tab main  
par mtext /GET MAIN c.1 [L1]  
par intext = "[INTEXT] [MTEXT]" // concatenate each row with the intext string  
LOOP END
```

```

// Remove punctuation marks from our text string - substitute "" (no chars) for identified marks
util.text.new /PAR INTEXT
util.text.replace /ANY - ""
util.text.replace /ANY . ""
util.text.replace /ANY , ""
util.text.replace /ANY ? ""
util.text.replace /ANY ! ""
util.text.replace /ANY " " " "
util.text.replace /ANY " " " "
// util.text.replace /ANY ' "" // Leave the single quote mark
util.text.replace /ANY - ""
util.text.replace /ANY ; ""
util.text.replace /ANY ' ""
util.text.replace /ANY ' ""
util.text.replace /ANY ) ""
util.text.replace /ANY ( ""

// PAR TEXTLENGTH = "[INTEXT]" // Par picked up from INPUT, parentheses because of spaces in the text
var.par TEXTLENGTH = /TEXT LENGTH "[INTEXT]" // How many characters in the string?

SECTION Find-Words-in-String
// -----

DELETE /TAB MAIN // Clear out MAIN

add /COL c.1 Word_Count INT
add /COL c.2 Word text
par counter = 1

LOOP L1 IS [TEXTLENGTH] > 0 // do this while there is text left to analyse

PAR WORD = "[INTEXT]"
var.par TEXTLENGTH = /TEXT LENGTH "[INTEXT]" // How many characters remain in the string?

```

```

PAR Rlen = [TEXTLENGTH]
var.par wl = /TEXT findstring " " = "[INTEXT]"           // Look for the next occurrence of a space

IF IS [WL] = 0

    var.par RLEN = /MATHS MINUS 2
    var.par INTEXT = /TEXT MID 1 [RLEN]

    LOOP CONTINUE

IF END

If is [WL] = -1
    PAR WORD = [INTEXT]           // At the end of the line ...no more spaces when -1 is returned
If else
    var.par WORD = /TEXT LEFT [WL]           // grab the next word [WL] tells us how many characters to grab off the left
If end

add /ROW 1           // add another row to MAIN and put the new word into the next row
set /TABVAR MAIN c.1 [COUNTER] [COUNTER]
set /TABVAR MAIN c.2 [COUNTER] [WORD]

IF IS [WL] = -1
LOOP BREAK

IF ELSE
var.par Rlen = /MATHS MINUS [WL]
var.par Rlen = /MATHS MINUS 1           // 'cos string positions start at 0
var.par INTEXT = /TEXT RIGHT [RLEN]
var.par counter = /GEN INCREMENT 1
IF END

LOOP END

```

SECTION Show-Results

// -----

REM We've finished the word finding, so group / count the words by no. of occurrences and display them

GROUP c.2 COUNT // Group by word to get a count of occurrences

SORT ASC c.2 // And put in alphabetic order

TAB.SHOW MAIN "Results in alphabetic order"

SORT DESC c.1 // And put in descending order of word count

TAB.SHOW MAIN "Results in word count order"

REM Present a Pie Chart

// Add a sequence and keep just the top 15

Add /COL c.3 Seq INT

col c.3 = csum 1

Mark c.3 > 15

Remove /CLEAN

COL c.2 = c.2 + " x" + c.1

GRAPH.CLEAR

GRAPH.FRAME "TITLE=Top 15 Words used"

GRAPH.PIE.COL c.1 c.2 // Col 1 is count

GRAPH.SHOW

Msg "Goodnight Sweet Prince/Princess"

EXIT "Goodbye Sweet Prince/Princess"