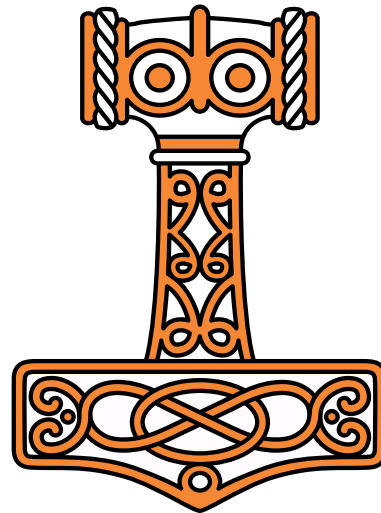
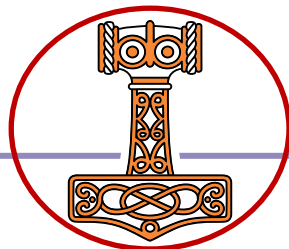
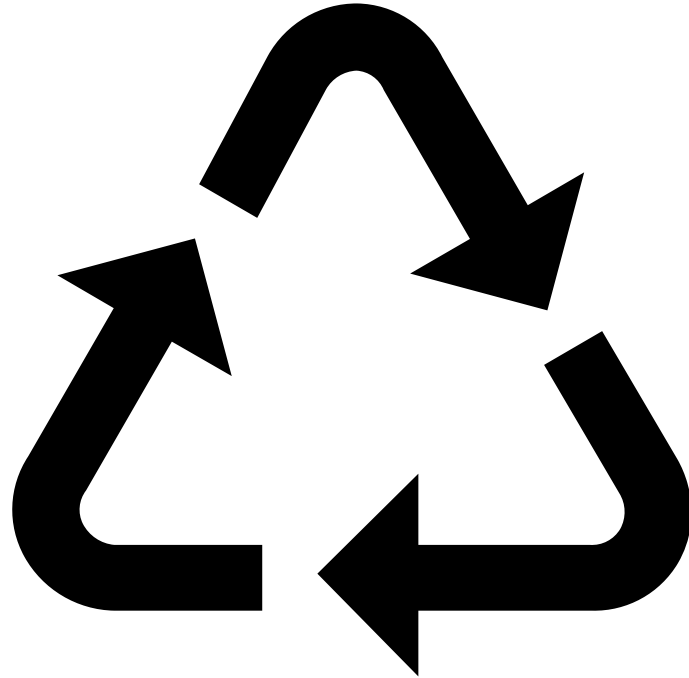




Working with Non-APL Data Sources

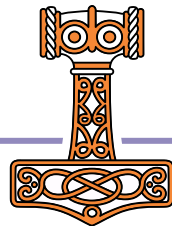
Richard Smith and Bjørn Christensen





Introduction

- Simple database, in two spreadsheets
- Various ways of importing the data
- A “Quick and dirty” way is to convert to CSV files
- Will use the CSV files throughout the presentation



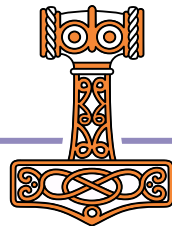
□ CSV: ~~Comma~~ Separated Values

- Text file containing records split into fields, using a comma as a field separator

```
Hats,1.2  
Scarves,1234
```

- Other delimiters are used – e.g. in Europe a semicolon is more usual

```
Hats;1,2  
Scarves;1234
```



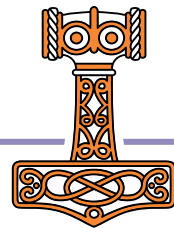
The “database”

Products

Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

Deliveries

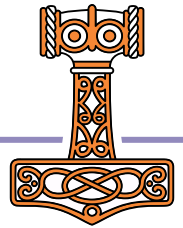
Date	Code	Quantity
2022-06-18 02:23:00	SSM	7
2022-06-18 02:23:00	SKM	4
2022-06-25 03:40:00	SSM	7
2022-06-25 03:40:00	SKM	4
2022-07-02 02:50:10	SSM	8
2022-07-02 02:50:10	SKM	3
2022-07-09 05:20:00	SSM	5
2022-07-09 05:20:00	SKM	1
2022-07-16 03:27:34	SSM	6
2022-07-16 03:27:34	SKM	2
2022-07-23 04:18:44	SSM	8
2022-07-23 04:18:44	SKM	4
2022-07-30 01:58:10	SSM	8
2022-07-30 01:58:10	SKM	3
2022-08-06 03:29:39	SSM	4
2022-08-06 03:29:39	SKM	1



Microsoft Excel interface showing a spreadsheet with data. The ribbon is set to 'Home'. The spreadsheet has columns A, B, and C, and rows 1 through 24. The data is as follows:

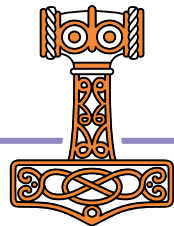
Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

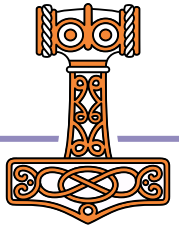
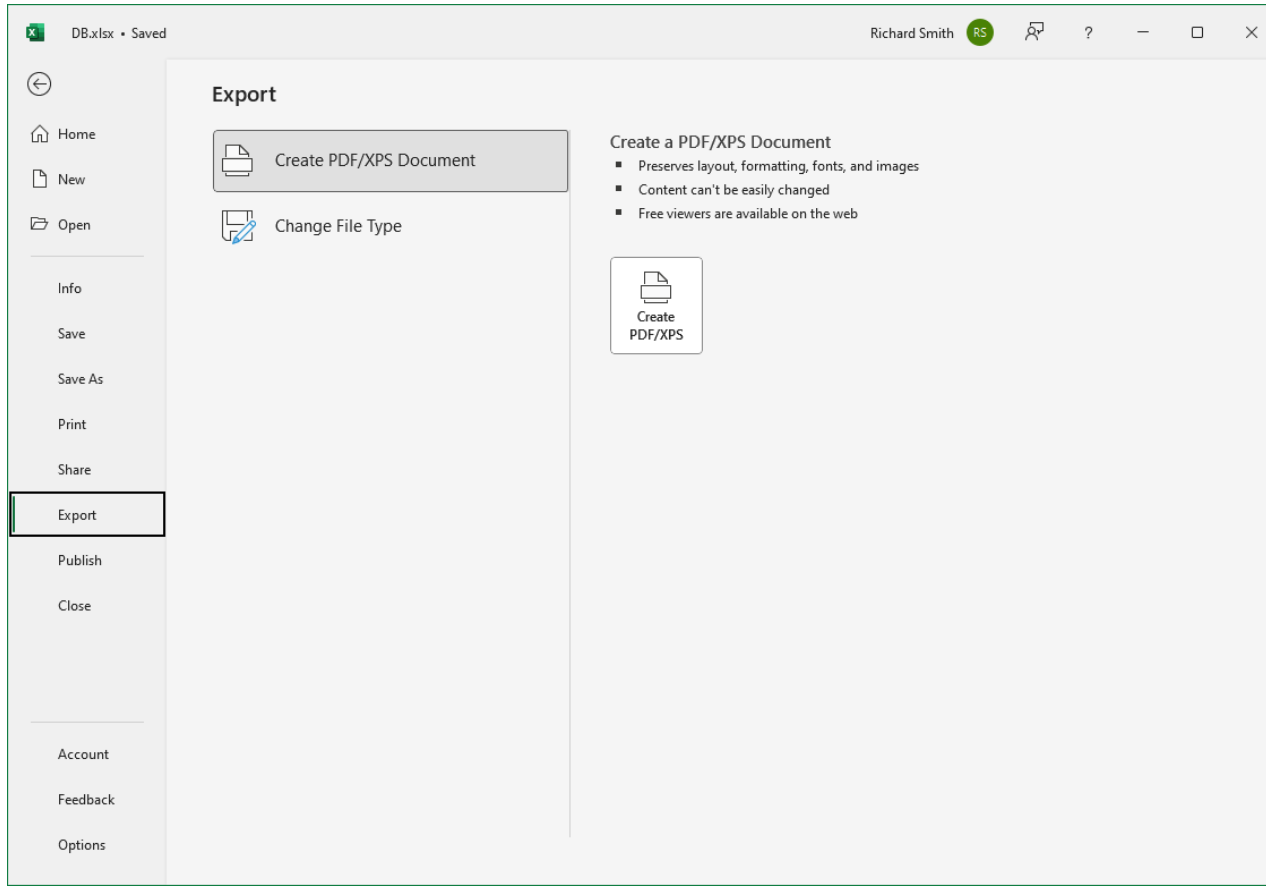
The status bar at the bottom indicates 'Ready' and 'Accessibility: Good to go'.

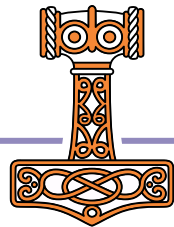
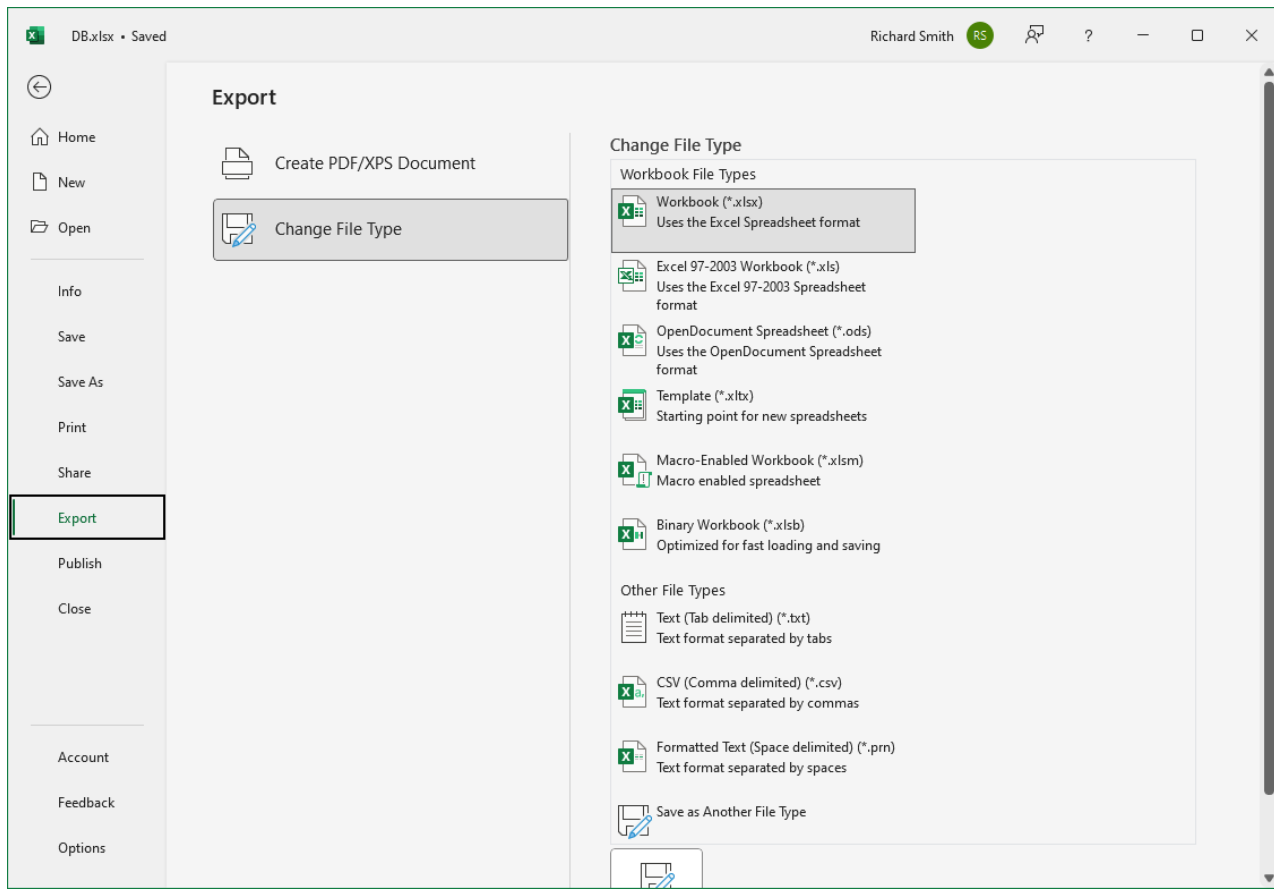


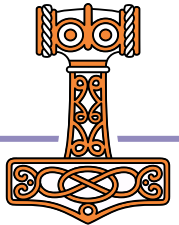
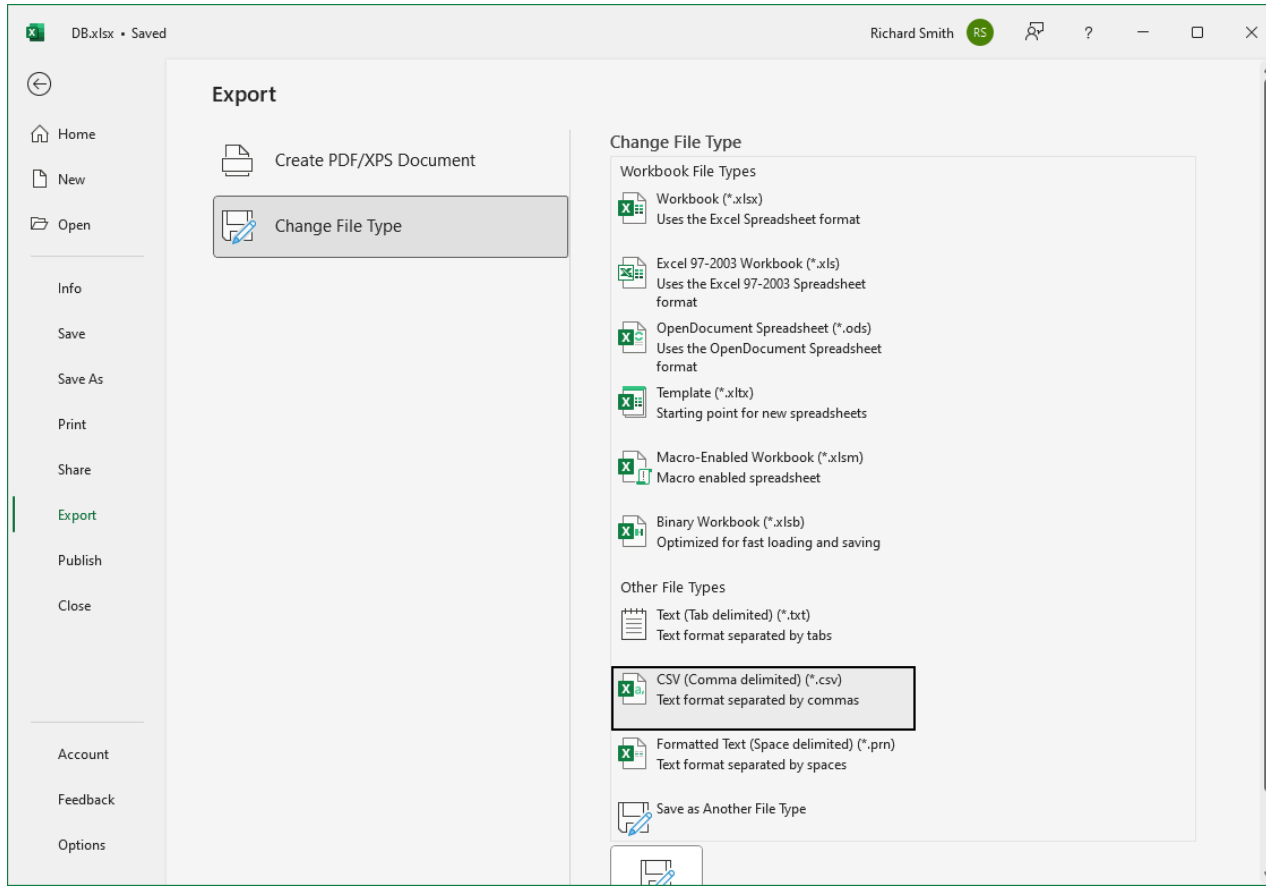
Microsoft Excel interface showing a spreadsheet with data. The ribbon is set to the Home tab. The spreadsheet displays a table with columns labeled Date, Code, and Quantity. The data is organized into rows, with the first row (row 1) containing headers. The data spans from row 2 to row 17. The status bar at the bottom indicates 'Ready' and 'Accessibility: Good to go'.

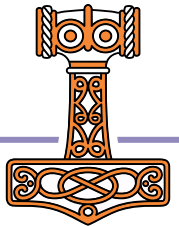
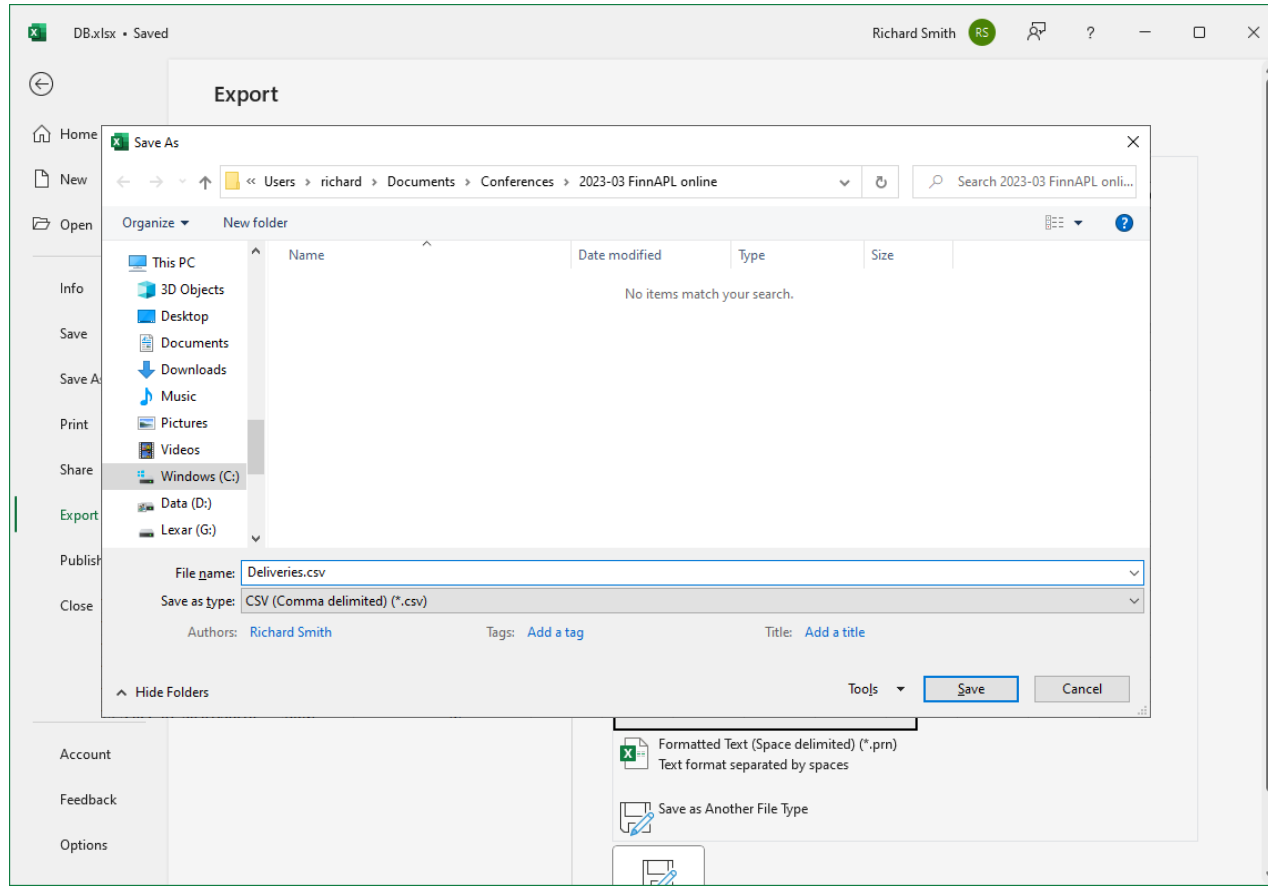
Date	Code	Quantity
2022-06-18 02:23:00	SSM	7
2022-06-18 02:23:00	SKM	4
2022-06-25 03:40:00	SSM	7
2022-06-25 03:40:00	SKM	4
2022-07-02 02:50:10	SSM	8
2022-07-02 02:50:10	SKM	3
2022-07-09 05:20:00	SSM	5
2022-07-09 05:20:00	SKM	1
2022-07-16 03:27:34	SSM	6
2022-07-16 03:27:34	SKM	2
2022-07-23 04:18:44	SSM	8
2022-07-23 04:18:44	SKM	4
2022-07-30 01:58:10	SSM	8
2022-07-30 01:58:10	SKM	3
2022-08-06 03:29:39	SSM	4
2022-08-06 03:29:39	SKM	1











AutoSave Off DB.xlsx • Saved

Search

Richard Smith

File Home Insert Draw Page Layout Formulas Data Review View Automate Add-ins Help Team

Comments Share

Undo Clipboard Font Alignment Number Styles Cells Editing Analysis

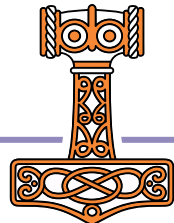
A19

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date	Code	Quantity												
2	2022-06-18 02:23:00	SSM	7												
3	2022-06-18 02:23:00	SKM	4												
4	2022-06-25 03:40:00	SSM	7												
5	2022-06-25 03:40:00	SKM	4												
6	2022-07-02 02:50:10	SSM	8												
7	2022-07-02 02:50:10	SKM	3												
8	2022-07-09 05:20:00	SSM	5												
9	2022-07-09 05:20:00	SKM	1												
10	2022-07-16 03:27:34	SSM	6												
11	2022-07-16 03:27:34	SKM	2												
12	2022-07-23 04:18:44	SSM	8												
13	2022-07-23 04:18:44	SKM	4												
14	2022-07-30 01:58:10	SSM	8												
15	2022-07-30 01:58:10	SKM	3												
16	2022-08-06 03:29:39	SSM	4												
17	2022-08-06 03:29:39	SKM	1												
18															
19															
20															
21															
22															
23															
24															

Products Deliveries

Ready Accessibility: Good to go

100%



AutoSave Off DB.xlsx • Saved Search Richard Smith

File Home Insert Draw Page Layout Formulas Data Review View Automate Add-ins Help Team Comments Share

Calibri 11 General Conditional Formatting Insert Sum Z A Analyze B I U A A Format as Table Delete

Deliveries.csv - Notepad

File Edit Format View Help

Date;Code;Quantity

2022-06-18 02:23:00;SSM;7

2022-06-18 02:23:00;SKM;4

2022-06-25 03:40:00;SSM;7

2022-06-25 03:40:00;SKM;4

2022-07-02 02:50:10;SSM;8

2022-07-02 02:50:10;SKM;3

2022-07-09 05:20:00;SSM;5

2022-07-09 05:20:00;SKM;1

2022-07-16 03:27:34;SSM;6

2022-07-16 03:27:34;SKM;2

2022-07-23 04:18:44;SSM;8

2022-07-23 04:18:44;SKM;4

2022-07-30 01:58:10;SSM;8

2022-07-30 01:58:10;SKM;3

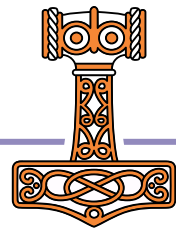
2022-08-06 03:29:39;SSM;4

2022-08-06 03:29:39;SKM;1

Ln 1, Col 1 100% Windows (CRLF) UTF-8

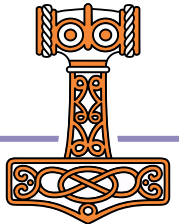
Products Deliveries

Ready Accessibility: Good to go



□CSV has *lots* of config options

Field separator characters, escape mechanism, decimal separator characters, thousands separator characters, redundant whitespace trimming, even or uneven length records, fixed or variable-width fields, quotation mark characters, output matrix format, specification of column datatypes, handling of missing fields, input source specification, separation of header record.

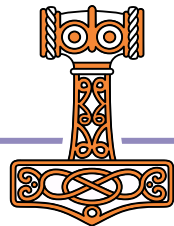


CSV has *lots* of config options

```
csv←CSV('Separator' ';' )('Decimal' ',' )
```

```
(csv['Invert' <invert_type>]
```

```
<filename> <encoding> <col_type> <header>
```

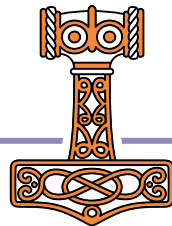


CSV has *lots* of config options

```
csv←CSV('Separator' ';' )('Decimal' ',' )
```

```
(csv['Invert' <invert_type>)
```

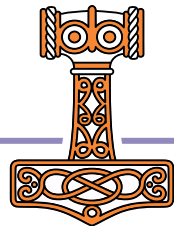
```
<filename> <encoding> <col_type> <header>
```



CSV has *lots* of config options

```
csv←CSV('Separator' ';' )('Decimal' ',' )
```

```
(csv['Invert' <invert_type>)  
  <filename> <encoding> <col_type> <header>
```

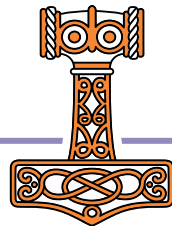


CSV has *lots* of config options

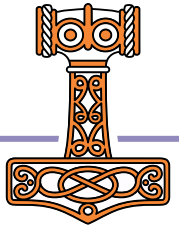
```
csv←CSV('Separator' ';' )('Decimal' ',' )
```

```
(csv['Invert' <invert_type>)
```

```
<filename> <encoding> <col_type> <header>
```



Demonstration



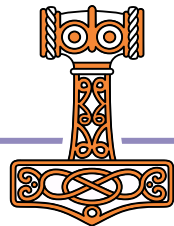
How much did I spend?

Products

Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

Deliveries

Date	Code	Quantity
2022-06-18 02:23:00	SSM	7
2022-06-18 02:23:00	SKM	4
2022-06-25 03:40:00	SSM	7
2022-06-25 03:40:00	SKM	4
2022-07-02 02:50:10	SSM	8
2022-07-02 02:50:10	SKM	3
2022-07-09 05:20:00	SSM	5
2022-07-09 05:20:00	SKM	1
2022-07-16 03:27:34	SSM	6
2022-07-16 03:27:34	SKM	2
2022-07-23 04:18:44	SSM	8
2022-07-23 04:18:44	SKM	4
2022-07-30 01:58:10	SSM	8
2022-07-30 01:58:10	SKM	3
2022-08-06 03:29:39	SSM	4
2022-08-06 03:29:39	SKM	1



How much did I spend?

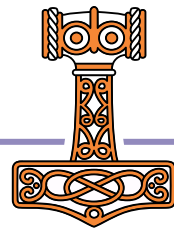
Products

Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

Deliveries

Date	Code	Quantity
2022-06-18 02:23:00	SSM	7
2022-06-18 02:23:00	SKM	4
2022-06-25 03:40:00	SSM	7
2022-06-25 03:40:00	SKM	4
2022-07-02 02:50:10	SSM	8
2022-07-02 02:50:10	SKM	3
2022-07-09 05:20:00	SSM	5
2022-07-09 05:20:00	SKM	1
2022-07-16 03:27:34	SSM	6
2022-07-16 03:27:34	SKM	2
2022-07-23 04:18:44	SSM	8
2022-07-23 04:18:44	SKM	4
2022-07-30 01:58:10	SSM	8
2022-07-30 01:58:10	SKM	3
2022-08-06 03:29:39	SSM	4
2022-08-06 03:29:39	SKM	1

$$7 \times 0.92 = 6.44$$



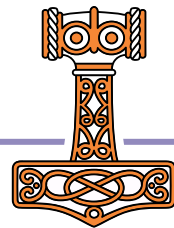
How much did I spend?

Products

Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

Deliveries

Date	Code	Quantity	
2022-06-18 02:23:00	SSM	7	$7 \times 0.92 = 6.44$
2022-06-18 02:23:00	SKM	4	$4 \times 0.91 = 3.64$
2022-06-25 03:40:00	SSM	7	etc...
2022-06-25 03:40:00	SKM	4	
2022-07-02 02:50:10	SSM	8	
2022-07-02 02:50:10	SKM	3	
2022-07-09 05:20:00	SSM	5	
2022-07-09 05:20:00	SKM	1	
2022-07-16 03:27:34	SSM	6	
2022-07-16 03:27:34	SKM	2	
2022-07-23 04:18:44	SSM	8	
2022-07-23 04:18:44	SKM	4	
2022-07-30 01:58:10	SSM	8	
2022-07-30 01:58:10	SKM	3	
2022-08-06 03:29:39	SSM	4	
2022-08-06 03:29:39	SKM	1	



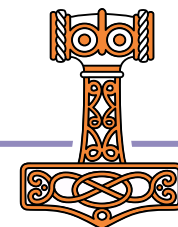
How much did I spend in July?

Products

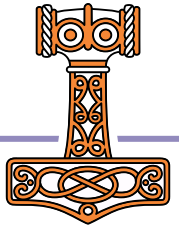
Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

Deliveries

Date	Code	Quantity	+/-
2022-06-18 02:23:00	SSM	7	$7 \times 0.92 = 6.44$
2022-06-18 02:23:00	SKM	4	$4 \times 0.91 = 3.64$
2022-06-25 03:40:00	SSM	7	etc...
2022-06-25 03:40:00	SKM	4	
2022-07-02 02:50:10	SSM	8	
2022-07-02 02:50:10	SKM	3	
2022-07-09 05:20:00	SSM	5	
2022-07-09 05:20:00	SKM	1	
2022-07-16 03:27:34	SSM	6	
2022-07-16 03:27:34	SKM	2	
2022-07-23 04:18:44	SSM	8	
2022-07-23 04:18:44	SKM	4	
2022-07-30 01:58:10	SSM	8	
2022-07-30 01:58:10	SKM	3	
2022-08-06 03:29:39	SSM	4	
2022-08-06 03:29:39	SKM	1	

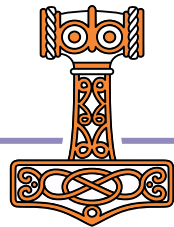


Demonstration



JSON

- Text containing structured data:
 - Numbers
 - Strings (character arrays)
 - Objects (namespaces)
 - Vectors
- JavaScript Object Notation



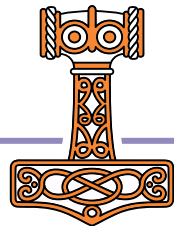
JSON

JSON 1.3 'Hello'

[1.3, "Hello"]

JSON '[1.3, "Hello"]'

1.3	Hello
-----	-------

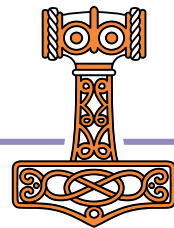


JSON

- Text containing structured data:

- Numbers
- Strings (character arrays)
- Objects (namespaces)
- Vectors
- Booleans, null
- Infinity, NaN

```
json5 ← JSON 'Dialect' 'JSON5'
```



Use monadic ⌈JSON carefully

```
⌈JSON ⌈A
```

```
DOMAIN ERROR: JSON import: invalid character
```

```
⌈JSON ⌈A
```

```
^
```

```
json_in←0∘⌈JSON
```

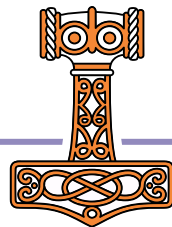
```
json_out←1∘⌈JSON
```

```
json_out ⌈A
```

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
⌈JSON 1.3 'Hello'  
[1.3,"Hello"]  
⌈JSON '[1.3,"Hello"]'
```

1.3	Hello
-----	-------



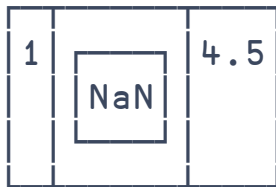
JSON types without APL equivalents

0 json5 'NaN'



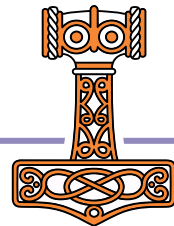
“Wrapper”

0 json5 '[1, NaN, 4.5]'

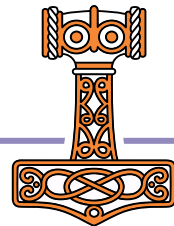
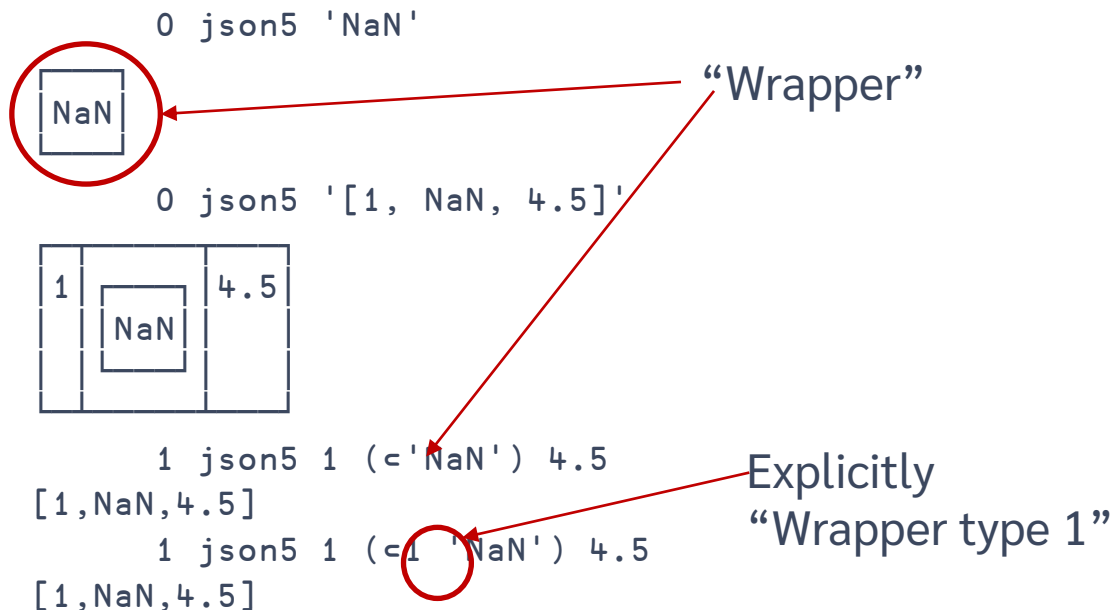


1 json5 1 (c 'NaN') 4.5

[1,NaN,4.5]

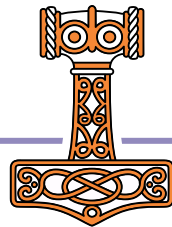


JSON types without APL equivalents



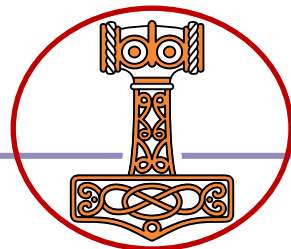
Tables

- JSON does not support matrices
- □JSON has an option to automatically split high-rank arrays



JSON-stat

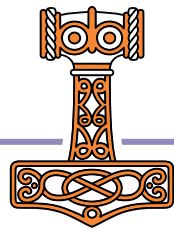
- ❖ Veli-Matti introduced me to JSON-stat at the Dyalog User Meeting last year
- ❖ What follows is *not* JSON-stat



JSON tables

- Tables exist outside of APL!
- Convention established for tables with:
 - Some columns
 - One header row
 - Some rows of data

Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89



A simple and common convention

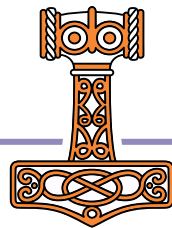
A table with c columns, 1 header row and r rows of data can be represented as:

An array of r objects, where each object consists of:

c items, where each item's name is what is in the column's header

And each item's value is the cell content

For example ...



Code	Description	Cost
FFM	Pasteurised milk 1pt	0.94
SSM	Semi-skimmed milk 1pt	0.92
SKM	Skimmed milk 1pt	0.91
EGG	Half-dozen eggs	1.89

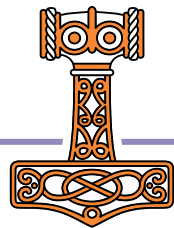


Matrix

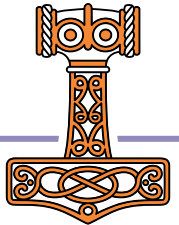
Vector of namespaces 

1 JSON ?

```
[
  {
    "Code": "FFM",
    "Description": "Pasteurised milk 1pt",
    "Cost": 0.94
  },
  {
    "Code": "SSM",
    "Description": "Semi-skimmed milk 1pt",
    "Cost": 0.92
  },
  {
    "Code": "SKM",
    "Description": "Skimmed milk 1pt",
    "Cost": 0.91
  },
  {
    "Code": "EGG",
    "Description": "Half-dozen eggs",
    "Cost": 1.89
  }
]
```



Demonstration



End of part 1

