DYALOG

Elsinore 2023

APL Functions to Import, Export and Process Data in Files

Richard Smith

Peter Mikkelsen

Setup

- Copy FILE*.txt from https://github.com/dyalog-training/2023-TP3
- Dyalog 19.0
  - 18.2 ok for all but one exercise
- In APL session, do:
  ```
  ]CD "/Path/To/Files"
  ```

Agenda

- **Text files**
- CSV files
- Portable file functions
- JSON
- XML

Aren’t text files really simple?

- Files just contain a stream of bytes, which are easily mapped to characters.
- … so long as everyone only needs a few distinct characters and glyphs.
- … and everyone agrees the line-ending style.
- … and everyone uses the same mapping.

History

- Pre-ASCII (inc ASCII 1963).
- ASCII: 7-bit:
  - Includes file control chars.
  - Includes DEL at 0x7F.
  - Common US-English chars.
    - UK: £ vs #.
- Extended ASCII: 8-bit:
  - Region-specific chars.
  - Different mappings 128→.
- EBCDIC: 8 bit:
  - Entirely different ordering.
  - Region-specific chars.
  - Different mappings.

History

- UNICODE:
  - UTF-16.
  - UTF-32.
  - UTF-16 with Surrogates.
  - Byte ordering and BOMs.
  - UTF-8.
- Line-Endings:
  - CR LF.
  - CR.
  - LF.
  - NEL (UNICODE).
  - ...
  - Last line?

Text files are not at all simple!

- Dyalog tries to make reading and writing them as simple as possible.
- Three “mystery” text files: FILE1.txt, FILE2.txt, FILE3.txt.

Demo and exercise 1

```
      ]CD "C:/Users/richard/Documents/Conferences/2023-10 User Meeting/Workshop/2023-TP3/"
C:\Users\richard\Documents\Conferences\2023-10 User Meeting\Workshop\2023-TP3
      t←'FILE1.txt' ⎕NTIE 0
      ⎕NREAD t 83 100 0
84 104 101 32 80 114 111 106 101 99 116 32 71 117 116 101 110 98 101 114 103 32 101 66 111 111 107 32 111 102 32 70
97 105 114 121 32 84 97 108 101 115 32 111 102 32 72 97 110 115 32 67 104 114 105 115 116 105 97 110 32 65 110 100
101 114 115 101 110 13 10 32 32 32 32 13 10 84 104 105 115 32 101 98 111 111 107 32 105 115 32 102 111 114 32 116
104 101 32 117
      ⎕NREAD t 80 100 0 ⍝ Classic use 82
The Project Gutenberg eBook of Fairy Tales of Hans Christian Andersen



This ebook is for the u
      ⎕NUNTIE t
      f←{t←⍵ ⎕NTIE 0 ⋄ r←⎕NREAD t 80 100 0 ⋄ _←⎕NUNTIE t ⋄ r}
      f 'FILE1.txt'
The Project Gutenberg eBook of Fairy Tales of Hans Christian Andersen



This ebook is for the u
      f 'FILE2.txt'
   P   r   o   j   e   c   t       G   u   t   e   n   b   e   r   g   '   s       H   a   n   s
      f 'FILE3.txt'
%@@@@ã
@×
 £@Ç¤£

  @
Â   @  @â£
¢@    @È  ¢@Á
 ¢
 %@@@@%ã  ¢@
    @ ¢@    @£
@¤¢
```

@ @ ..
A Exercise 1: see how well your OS handles them.

⎕NGET and ⎕NPUT

- Read and write text files
  - Powerful, but simple in almost all cases.

⎕NGET and ⎕NPUT

- ⎕NGET *filename*
  - Deduces the encoding used.
  - Returns a 3-element array:
    - The text, with normalised line-endings.
    - The deduced encoding.
    - The line-ending type.

⎕NGET and ⎕NPUT

- Optional: encoding in left argument.
- Optional: return text as vector of character vectors.

Demo and Exercise 2

```
      d e l←⎕NGET 'FILE1.txt'
      ⍴d
2015345
      )ed d
      e
UTF-8-NOBOM
      l
13 10
      d e l←'UTF-8'⎕NGET 'FILE1.txt' 1
      ⍴d
38041
      ⍝ Exercise 2: try this for files FILE2.txt and FILE3.txt
```

```
      ⍝ Exercise 2 "solution"
      d e l←⎕NGET 'FILE2.txt'
      )ed d
      e
UTF-32BE-NOBOM
      l
13 10
      d e l←⎕NGET 'FILE3.txt'
TRANSLATION ERROR: Unable to decode the file
      d e l←⎕NGET'FILE3.txt'
            ^
            ⍝ Hmmm....
      key←⎕U2338 ⍝ Classic
SYNTAX ERROR: Invalid System Name: "⎕U2338"
      key←⎕U2338 ⍝ Classic
          ^
      t←'FILE1.txt' ⎕NTIE 0 ⍝ NB - this is FILE1.txt
      d←⎕NREAD t 83 ¯1
      ⎕NUNTIE t
      d2←256|d
      {⍺(≠⍵)}⌸d2
 84   5713
104 114245
101 205578
 32 343825
 80    750
114  88905
111 114359
106   1295
 99  31310
116 139986
 71   1021
117  40536
110 101991
 98  22902
103  31989
```

66	1359
107	12985
102	32042
70	569
97	121451
105	89059
121	27847
108	68046
115	93588
72	2334
67	867
65	2514
100	74588
13	38041
10	38041
119	41831
85	179
83	2057
109	33516
112	23027
118	13031
46	16195
89	746
44	29824
45	5109
76	711
73	5787
58	330
82	679
78	1065
56	15
50	13
48	23
91	5
35	1
55	8

```
 93      5
 77    866
 74    403
 52     13
 49     73
 69   1204
 42     22
 79   1127
 75    328
 68    797
 87   1394
 39   3020
120   1330
 81     47
 33   1531
 59   5357
 34  10579
113   1257
 63   1145
122    610
 86    239
 90     11
 40     32
 41     32
 88     21
 57      9
 38      4
 53     12
 54      9
 51     12
 47      6
 37      1
 36      2
     10↑{ω[⍣ω[;2];]}{α(≠ω)}⍤d2
 32 343825
101 205578
```

```
116 139986
 97 121451
111 114359
104 114245
110 101991
115  93588
105  89059
114  88905
      10↑{ω[⍒ω[;2];]}{(⎕UCS α)(≠ω)}⌸d2
  343825
e 205578
t 139986
a 121451
o 114359
h 114245
n 101991
s  93588
i  89059
r  88905
      t←'FILE3.txt' ⎕NTIE 0
      d←⎕NREAD t 83 ¯1
      ⎕NUNTIE t
      d2←256|d
      10↑{ω[⍒ω[;2];]}{α(≠ω)}⌸d2
 64 37879
133 22044
163 15272
129 12631
150 12128
136 11719
149 10786
137 10158
162 10093
153  9735
      ]open https://en.wikipedia.org/wiki/EBCDIC
https://en.wikipedia.org/wiki/EBCDIC
```

⎕NGET and ⎕NPUT

- Optional: encoding in left argument.
- 256-element numeric:
  - Unicode characters for byte values 0..255

Demo and Exercise 3

```
      d e l←⎕NGET 'FILE4.txt'
      )ed d
      ns←0 ⎕JSON d
      ↑ns[⍳⍴ns].(Name Note)
 ISO/IEC 8859 Part 1   Latin-1 Western European
 ISO/IEC 8859 Part 2   Latin-2 Central European
 ISO/IEC 8859 Part 3   Latin-3 South European
 ISO/IEC 8859 Part 4   Latin-4 North European
 ISO/IEC 8859 Part 5   Latin/Cyrillic
 ISO/IEC 8859 Part 6   Latin/Arabic
 ISO/IEC 8859 Part 7   Latin/Greek
 ISO/IEC 8859 Part 8   Latin/Hebrew
 ISO/IEC 8859 Part 9   Latin-5 Turkish
 ISO/IEC 8859 Part 10  Latin-6 Nordic
 ISO/IEC 8859 Part 11  Latin/Thai
 ISO/IEC 8859 Part 13  Latin-7 Baltic Rim
 ISO/IEC 8859 Part 14  Latin-8 Celtic
 ISO/IEC 8859 Part 15  Latin-9
 ISO/IEC 8859 Part 16  Latin-10 South-Eastern European
 Windows-874           Thai
 Windows-1250          Central and Eastern Europe
 Windows-1251          Cyrillic
 Windows-1252          Western Europe
 Windows-1253          Greek
 Windows-1254          Turkish
 Windows-1255          Hebrew
 Windows-1256          Arabic
 Windows-1257          Eastern Europe
 Windows-1258          Vietnamese
 IBM037                EBCDIC Latin-1
 IBM500                EBCDIC International Latin-1
 IBM875                EBCDIC Greek
 IBM1026               EBCDIC Latin-5 Turkish
      ns[1].Unicode
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
```

42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201
202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230
231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
A Exercise 3: Now use NGET to read FILE3.txt

```
      ⍝ Exercise 3 solution
      ns[27].(Name Note)
 IBM500  EBCDIC International Latin-1
      ns[27].Unicode
0 1 2 3 156 9 134 127 151 141 142 11 12 13 14 15 16 17 18 19 157 133 8 135 24 25 146 143 28 29 30 31 128 129 130 131
132 10 23 27 136 137 138 139 140 5 6 7 144 145 22 147 148 149 150 4 152 153 154 155 20 21 158 26 32 160 226 228 224
225 227 229 231 241 91 46 60 40 43 33 38 233 234 235 232 237 238 239 236 223 93 36 42 41 59 94 45 47 194 196 192 193
195 197 199 209 166 44 37 95 62 63 248 201 202 203 200 205 206 207 204 96 58 35 64 39 61 34 216 97 98 99 100 101 102
103 104 105 171 187 240 253 254 177 176 106 107 108 109 110 111 112 113 114 170 186 230 184 198 164 181 126 115 116
117 118 119 120 121 122 161 191 208 221 222 174 162 163 165 183 169 167 182 188 189 190 172 124 175 168 180 215 123
65 66 67 68 69 70 71 72 73 173 244 246 242 243 245 125 74 75 76 77 78 79 80 81 82 185 251 252 249 250 255 92 247 83
84 85 86 87 88 89 90 178 212 214 210 211 213 48 49 50 51 52 53 54 55 56 57 179 219 220 217 218 159
      d e l←(ns[27].Unicode) ⎕NGET 'FILE3.txt'
      )ed d
```

Agenda

- Text files
- **CSV files**
- Portable file functions
- JSON
- XML

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
```

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.


NESPRESSO

The “database”

Products

Code	Description	Cost	Vol
7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Choc Fudge	0.65	230
7211	Master Origins Columbia	0.69	230
7216	Altissio	0.48	40
7218	Di[illegible]litto	0.4[illegible]	[illegible]0

Orders

Date	Code	Quantity
12/09/2023	7216	50
12/09/2023	7218	30
07/08/2023	7225	50
21/06/2023	7197	30
21/06/2023	7201	10
21[illegible]	[illegible]	[illegible]34

Demo 4

The demo is in FILE7.txt

```
      fn_products←'FILE5.txt'
      fn_orders←'FILE6.txt'
      ⎕NGET fn_products
Code;Description;Cost;Vol
7197;Stormio;0,62;230
7201;Melozio;0,62;230
7209;Barista Creations Chocolate Fudge;0,65;230
7211;Master Origins Colombia;0,69;230
7216;Altissio;0,48;40
7218;Diavolitto;0,48;40
7219;Dolce;0,55;80
7225;Scuro;0,55;80
7229;Barista Creations Bianco Doppio;0,56;80
7234;Melozio Decaf;0,64;230
7244;Altissio Decaf;0,50;40

      208↑⎕NGET fn_orders
Date;Code;Quantity
12/09/2023;7216;50
12/09/2023;7218;30
07/08/2023;7225;50
21/06/2023;7197;30
21/06/2023;7201;10
21/06/2023;7234;10
21/06/2023;7225;50
21/06/2023;7216;50
21/06/2023;7211;50
19/06/2023;7197;30
      ⍴ ⎕CSV fn_products 'ASCII'
      ⎕CSV fn_products
DOMAIN ERROR: Invalid number of fields in record 2 (⎕IO=1)
      ⎕CSV fn_products
      ^
      (⎕CSV⍠'Separator' ';') fn_products ⍴ Classic use ⎕OPT
 Code  Description                    Cost  Vol
 7197  Stormio                        0,62  230
```

```
 7201  Melozio                            0,62  230
 7209  Barista Creations Chocolate Fudge  0,65  230
 7211  Master Origins Colombia            0,69  230
 7216  Altissio                           0,48  40
 7218  Diavolitto                         0,48  40
 7219  Dolce                              0,55  80
 7225  Scuro                              0,55  80
 7229  Barista Creations Bianco Doppio    0,56  80
 7234  Melozio Decaf                      0,64  230
 7244  Altissio Decaf                     0,50  40
      □DR¨(□CSV□'Separator' ';') fn_products
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
80 80 80 80
      csv←□CSV□('Separator' ';')('Decimal' ',')
      csv fn_products '' 4
 Code  Description                        Cost  Vol
 7197  Stormio                            0.62  230
 7201  Melozio                            0.62  230
 7209  Barista Creations Chocolate Fudge  0.65  230
 7211  Master Origins Colombia            0.69  230
 7216  Altissio                           0.48   40
 7218  Diavolitto                         0.48   40
 7219  Dolce                              0.55   80
 7225  Scuro                              0.55   80
 7229  Barista Creations Bianco Doppio    0.56   80
 7234  Melozio Decaf                      0.64  230
```

```
 7244  Altissio Decaf                      0.5    40
      ]DR¨csv fn_products '' 4
 80 80  80  80
163 80 645 163
163 80 645 163
163 80 645 163
163 80 645 163
163 80 645  83
163 80 645  83
163 80 645  83
163 80 645  83
163 80 645  83
163 80 645 163
163 80 645  83
      ]box on -style=min
Was OFF -style=min
      (csv⍠'Invert' 1) fn_products
```

Code	Description	Cost	Vol
7197	Stormio	0,62	230
7201	Melozio	0,62	230
7209	Barista Creations Chocolate Fudge	0,65	230
7211	Master Origins Colombia	0,69	230
7216	Altissio	0,48	40
7218	Diavolitto	0,48	40
7219	Dolce	0,55	80
7225	Scuro	0,55	80
7229	Barista Creations Bianco Doppio	0,56	80
7234	Melozio Decaf	0,64	230
7244	Altissio Decaf	0,50	40

```
      (csv⍠'Invert' 1) fn_products '' (2 1 2 2) 1
```

7197 7201 7209 7211 7216 7218 7219 7225 7229 7234 7244	Stormio Melozio Barista Creations Chocolate Fudge Master Origins Colombia Altissio Diavolitto Dolce	0.62 0.62 0.65 0.69 0.48 0.48 0.55 0.55 0.56 0.64 0.5	230 230 230 230 40 40 80 80 80 230 40

Code	Description	Cost	Vol

```
                                       Scuro
                                       Barista Creations Bianco Doppio
                                       Melozio Decaf
                                       Altissio Decaf

      ⊃(csv⎕'Invert' 1) fn_products '' (2 1 2 2) 1
7197 7201 7209 7211 7216 7218 7219 7225 7229 7234 7244 | Stormio                           | 0.62 0.62 0.65 0.69 0.48 0.48 0.55 0.55 0.56 0.64 0.5 | 230 230 230 230 40 40 80 80 80 230 40
                                                       | Melozio                           |
                                                       | Barista Creations Chocolate Fudge |
                                                       | Master Origins Colombia           |
                                                       | Altissio                          |
                                                       | Diavolitto                        |
                                                       | Dolce                             |
                                                       | Scuro                             |
                                                       | Barista Creations Bianco Doppio   |
                                                       | Melozio Decaf                     |
                                                       | Altissio Decaf                    |
      (csv⎕'Invert' 2) fn_products '' (2 1 2 2) 1
7197 7201 7209 7211 7216 7218 7219 7225 7229 7234 7244 | Stormio | Melozio | Barista Creations Chocolate Fudge | Master Origins Colombia | Altissio | Diavolitto | Dolce | Scuro | Barista Creations Bianco Doppio | Melozio Decaf | Altissio Decaf | 0.62 0.62 0.65 0.69 0.48 0.48 0.55 0.55 0.56 0.64 0.5 | 230 230 230 230 40 40 80 80 80 230 40 || Code | Description | Cost | Vol
      ⊃(csv⎕'Invert' 2) fn_products '' (2 1 2 2) 1
7197 7201 7209 7211 7216 7218 7219 7225 7229 7234 7244 | Stormio | Melozio | Barista Creations Chocolate Fudge | Master Origins Colombia | Altissio | Diavolitto | Dolce | Scuro | Barista Creations Bianco Doppio | Melozio Decaf | Altissio Decaf | 0.62 0.62 0.65 0.69 0.48 0.48 0.55 0.55 0.56 0.64 0.5 | 230 230 230 230 40 40 80 80 80 230 40
      p_code desc cost vol←⊃(csv⎕'Invert' 2) fn_products '' (2 1 2 2) 1
      p_code
7197 7201 7209 7211 7216 7218 7219 7225 7229 7234 7244
      desc
| Stormio | Melozio | Barista Creations Chocolate Fudge | Master Origins Colombia | Altissio | Diavolitto | Dolce | Scuro | Barista Creations Bianco Doppio | Melozio Decaf | Altissio Decaf |
      cost
0.62 0.62 0.65 0.69 0.48 0.48 0.55 0.55 0.56 0.64 0.5
      vol
230 230 230 230 40 40 80 80 80 230 40
      date o_code qty←⊃(csv⎕'Invert' 2) fn_orders '' (1 2 2) 1
      date
| 12/09/2023 | 12/09/2023 | 07/08/2023 | 21/06/2023 | 21/06/2023 | 21/06/2023 | 21/06/2023 | 21/06/2023 | 21/06/2023 | 19/06/2023 | 19/06/2023 | 19/06/2023 | 19/06/2023 | 12/04/2023 | 12/04/2023 | 12/04/2023 | 12/04/2023 | 12/04/2023 | 12/04/2023 | 23/11/2023 | 23/11/2023 | 23/11/2023 | 23/11/2023 | 23/11/2023 | 23/11/2023 | 23/11/2023 |
      o_code
7216 7218 7225 7197 7201 7234 7225 7216 7211 7197 7218 7219 7229 7234 7201 7244 7219 7218 7229 7209 7197 7201 7234
7244 7218 7219
      qty
50 30 50 30 10 10 50 50 50 30 20 50 20 50 20 30 40 20 40 10 10 50 50 40 10 30
      ]box off
Was ON
```

```
      ⍝ ⎕CSV <f1>
      ]open https://help.dyalog.com
https://help.dyalog.com
```

Exercise 4

Import FILE5.txt (products) & FILE6.txt (orders)

(see FILE7.txt)

Then:

Deduce some interesting things, such as:

- Number of capsules in each order.
- Total spend.
- Most popular variety.

Exercise 4 walk-through

```
      ]box on -style=min
Was OFF -style=min
      date o_code qty←⊃(csv⍠'Invert' 2) fn_orders '' (1 2 2) 1
      p_code desc cost vol←⊃(csv⍠'Invert' 2) fn_products '' (2 1 2 2) 1
      ⍝ Number of capsules in each order
      date
```

12/09/2023	12/09/2023	07/08/2023	21/06/2023	21/06/2023	21/06/2023	21/06/2023	21/06/2023	21/06/2023	19/06/2023	19/06/2023	19/06/2023	19/06/2023	12/04/2023	12/04/2023	12/04/2023	12/04/2023	12/04/2023	12/04/2023	23/11/2023	23/11/2023	23/11/2023	23/11/2023	23/11/2023	23/11/2023	23/11/2023

```
      ⍸←i←{⊂⍵}⌸date              ⍝ Indices into orders table, grouping by date
```

1 2	3	4 5 6 7 8 9	10 11 12 13	14 15 16 17 18 19	20 21 22 23 24 25 26

```
      {date[⍵]}¨i                ⍝ See!
```

12/09/2023 12/09/2023	07/08/2023	21/06/2023 21/06/2023 21/06/2023 21/06/2023 21/06/2023 21/06/2023	19/06/2023 19/06/2023 19/06/2023 19/06/2023	12/04/2023 12/04/2023 12/04/2023 12/04/2023 12/04/2023 12/04/2023	23/11/2023 23/11/2023 23/11/2023 23/11/2023 23/11/2023 23/11/2023 23/11/2023

```
      {qty[⍵]}¨i                 ⍝ Corresponding quantities
```

50 30	50	30 10 10 50 50 50	30 20 50 20	50 20 30 40 20 40	10 10 50 50 40 10 30

```
      {+/qty[⍵]}¨i               ⍝ Summed
80 50 200 120 200 200
      ↑{(date[⊃⍵]),+/qty[⍵]}¨i   ⍝ Order dates and capsule counts
```

12/09/2023	80
07/08/2023	50
21/06/2023	200
19/06/2023	120
12/04/2023	200
23/11/2023	200

```
      ⍝ Total spend
      ⌹↑o_code qty               ⍝ Codes ordered and quantities - no costs!
7216 50
```

```
7218 30
7225 50
7197 30
7201 10
7234 10
7225 50
7216 50
7211 50
7197 30
7218 20
7219 50
7229 20
7234 50
7201 20
7244 30
7219 40
7218 20
7229 40
7209 10
7197 10
7201 50
7234 50
7244 40
7218 10
7219 30
      ⍞←x←p_code ⍳ o_code       ⍝ Index of each code to matching code in product table
5 6 8 1 2 10 8 5 4 1 6 7 9 10 2 11 7 6 9 3 1 2 10 11 6 7
      ]box off
Was ON
      ⍉↑desc[x] qty             ⍝ Names instead of codes using that index
 Altissio                        50
 Diavolitto                      30
 Scuro                           50
 Stormio                         30
 Melozio                         10
 Melozio Decaf                   10
```

```
 Scuro                              50
 Altissio                           50
 Master Origins Colombia            50
 Stormio                            30
 Diavolitto                         20
 Dolce                              50
 Barista Creations Bianco Doppio    20
 Melozio Decaf                      50
 Melozio                            20
 Altissio Decaf                     30
 Dolce                              40
 Diavolitto                         20
 Barista Creations Bianco Doppio    40
 Barista Creations Chocolate Fudge  10
 Stormio                            10
 Melozio                            50
 Melozio Decaf                      50
 Altissio Decaf                     40
 Diavolitto                         10
 Dolce                              30
      ]box on -style=min
Was OFF -style=min
      cost[x]                     ⍝ Costs corresponding to codes in orders table
0.48 0.48 0.55 0.62 0.62 0.64 0.55 0.48 0.69 0.62 0.48 0.55 0.56 0.64 0.62 0.5 0.55 0.48 0.56 0.65 0.62 0.62 0.64
0.5 0.48 0.55
      cost[x]×qty                 ⍝ Cost of each order line
24 14.4 27.5 18.6 6.2 6.4 27.5 24 34.5 18.6 9.6 27.5 11.2 32 12.4 15 22 9.6 22.4 6.5 6.2 31 32 20 4.8 16.5
      +/cost[x]×qty               ⍝ Total cost of all arders
480.4
      ⍝ Most popular variety
      ⎕←i←{⊂⍵}⌸o_code             ⍝ Indices into orders table, grouping by code
┌───┬───────────┬───┬───────┬───────┬───────┬─┬────────┬─────┬─────┬──┐
│1 8│2 11 18 25│3 7│4 10 21│5 15 22│6 14 23│9│12 17 26│13 19│16 24│20│
└───┴───────────┴───┴───────┴───────┴───────┴─┴────────┴─────┴─────┴──┘
```

```
      {o_code[⍵]}¨i               ⍝ See!
```

7216 7216	7218 7218 7218 7218	7225 7225	7197 7197 7197	7201 7201 7201	7234 7234 7234	7211	7219 7219 7219	7229 7229	7244 7244	7209

```
      {qty[⍵]}¨i                  ⍝ Corresponding quantities
```

50 50	30 20 20 10	50 50	30 30 10	10 20 50	10 50 50	50	50 40 30	20 40	30 40	10

```
      {+/qty[⍵]}¨i                ⍝ Summed
100 80 100 70 80 110 50 120 60 70 10
      ⎕←m←⊃⍒{+/qty[⍵]}¨i          ⍝ Index of maximum sum
8
      i[m]                        ⍝ Indices into orders table of corresponding orders
```

12 17 26

```
      ⎕←c←o_code[⊃⊃i[m]]          ⍝ Most popular code
7219
      p_code ⍳ c                  ⍝ Index into products table of that code
7
      desc[p_code ⍳ c]            ⍝ Most popular variety
```

Dolce

```
      ]box off
Was ON
```

Agenda

- Text files
- CSV files
- **Portable file functions**
- JSON
- XML

Demo and exercise 5

```
      ⎕MKDIR 'subdir'
      +'subdir' (⎕NCOPY⍠1) '*.TXT'
9
      +'subdir' (⎕NCOPY⍠1) '*.TXT'
FILE NAME ERROR: subdir/FILE1.txt: Already exists
      +'subdir'(⎕NCOPY⍠1)'*.TXT'
               ^
      +'subdir' (⎕NCOPY⍠('Wildcard' 1)('IfExists' 'Skip')) '*.TXT'
0
      +'subdir' (⎕NCOPY⍠('Wildcard' 1)('IfExists' 'Replace')) '*.TXT'
9
      +'subdir' (⎕NCOPY⍠('Wildcard' 1)('IfExists' 'ReplaceIfNewer')) '*.TXT'
0
      ↑¨(⎕NINFO⍠('Wildcard' 1)('Recurse' 1))'*.TXT'
 FILE1.txt
 FILE2.txt
 FILE3.txt
 FILE4.txt
 FILE5.txt
 FILE6.txt
 FILE7.txt
 FILE8.txt
 FILE9.txt
 subdir/FILE1.txt
 subdir/FILE2.txt
 subdir/FILE3.txt
 subdir/FILE4.txt
 subdir/FILE5.txt
 subdir/FILE6.txt
 subdir/FILE7.txt
 subdir/FILE8.txt
 subdir/FILE9.txt
      ↑¨0 3(⎕NINFO⍠('Wildcard' 1)('Recurse' 1)) 'FILE1.TXT'
 FILE1.txt         2023  9 12 10 53 36 193
 subdir/FILE1.txt  2023 10  9 11 18 11 221
      +'subdir'(⎕NCOPY⍠('Wildcard' 1)('IfExists' 'Replace')('PreserveAttributes' 1)) '*.TXT'
```

```
9
      ↑¨0 3(⎕NINFO⍠('Wildcard' 1)('Recurse' 1)) 'FILE1.TXT'
 FILE1.txt         2023 9 12 10 53 36 193
 subdir/FILE1.txt  2023 9 12 10 53 36 193
      ⍝ Exercise 5 ...
      1 ⎕MKDIR 'subdir'
      ncopy←⎕NCOPY⍠('Wildcard' 1)('IfExists' 'Replace')
      +'subdir' ncopy '*.TXT'
9
      ⍝ Wouldn't it be nice to see the progress as the files are copied?
      cb←{⎕←⎕JSON ⍵ ⋄ 1}
      'subdir' (ncopy⍠'ProgressCallback' 'cb') '*.TXT'
["⎕NCOPY","Start",{"Data":[],"Last":[],"Limit":0,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE1.txt"],"Limit":1,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE2.txt"],"Limit":2,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE3.txt"],"Limit":3,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE4.txt"],"Limit":4,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE5.txt"],"Limit":5,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE6.txt"],"Limit":6,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE7.txt"],"Limit":7,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE8.txt"],"Limit":8,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Scan",{"Data":[],"Last":["FILE9.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":0}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE1.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":1}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE2.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":2}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE3.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":3}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE4.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":4}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE5.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":5}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE6.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":6}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE7.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":7}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE8.txt"],"Limit":9,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":8}]
["⎕NCOPY","Progress",{"Data":[],"Last":["FILE9.txt"],"Limit":10,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":9}]
["⎕NCOPY","Done",{"Data":[],"Last":[],"Limit":10,"Options":{"Delay":0,"LastFileCount":1,"ScanFirst":1,"Skip":0},"Progress":10}]
      ⍝ Use the new 19.0 "callback" feature to display the names of files as they are copied.
      ⍝ Exercise 5 solution
      ⎕VR 'cb1'
     ∇ ret←cb1(Function Event Info)
[1]    ret←1
[2]    →(Event≠'Progress')/0
[3]    ⎕←⊃Info.Last
     ∇
```

```
      'subdir' (ncopy⍠'ProgressCallback' 'cb1') '*.TXT'
FILE1.txt
FILE2.txt
FILE3.txt
FILE4.txt
FILE5.txt
FILE6.txt
FILE7.txt
FILE8.txt
FILE9.txt
      ⍝ Wouldn't it be nice to have a list of files that were copied after the copy finished?
      ⎕VR 'cb2'
     ∇ ret←ns cb2(Function Event Info)
[1]    :Select Event
[2]    :Case 'Start'
[3]        ns.nl←0
[4]    :CaseList 'Progress' 'Done'
[5]        ns.nl,←Info.Last
[6]    :EndSelect
[7]    ret←1
     ∇
      ns←⎕NS''
      'subdir'(ncopy⍠'ProgressCallback'('cb2' ns))'*.TXT'
      ns.nl
 FILE1.txt  FILE2.txt  FILE3.txt  FILE4.txt  FILE5.txt  FILE6.txt  FILE7.txt  FILE8.txt  FILE9.txt
```

Agenda

- Text files
- CSV files
- Portable file functions
- **JSON**
- XML

JSON

- Text containing structured data:
  - Numbers.
  - Strings (character arrays).
  - Objects (namespaces).
  - Vectors.
- **J**ava**S**cript **O**bject **N**otation.

Demo 6

```
      ]box on -style=min
Was OFF -style=min
      jsontext
{
  "a": 1,
  "b": [
    2,
    "Hello"
  ],
  "c": {
    "x": 4
  },
  "d-e": 5
}
      v←⍬←⎕JSON jsontext
#.[JSON object]
      v.b
┌─┬─────┐
│2│Hello│
└─┴─────┘
      v.c
#.[JSON object].[JSON object]
      v.c.x
4
      ⎕JSON v
{"a":1,"b":[2,"Hello"],"c":{"x":4},"d-e":5}
      (⎕JSON⍠'Compact' 0) v
{
  "a": 1,
  "b": [
    2,
    "Hello"
  ],
  "c": {
    "x": 4
  },
```

```
  "d-e": 5
}
      v.⎕NL-⍳9
┌─┬─┬─┬─────┐
│a│b│c│⍙d⍙45⍙e│
└─┴─┴─┴─────┘
      1(7162⌶)¨v.⎕NL-⍳9
┌─┬─┬─┬───┐
│a│b│c│d-e│
└─┴─┴─┴───┘
      0 (7162⌶)¨'a' 'b' 'c' 'd-e'
┌─┬─┬─┬─────┐
│a│b│c│⍙d⍙45⍙e│
└─┴─┴─┴─────┘
      ⍪'v.',0 (7162⌶) 'd-e'
5
      json5←⎕JSON⍠'Dialect' 'JSON5'
      (json5⍠'Compact' 0) v
{
  a: 1,
  b: [
    2,
    "Hello",
  ],
  c: {
    x: 4,
  },
  "d-e": 5,
}
      json5 (json5⍠'Compact' 0) v
#.[JSON object]
      ⎕JSON 2 2 ⍴ 1
DOMAIN ERROR: JSON export: the right argument cannot be converted
      ⎕JSON 2 2⍴1
      ^
      csv fn_products
```

Code	Description	Cost	Vol
7197	Stormio	0,62	230
7201	Melozio	0,62	230
7209	Barista Creations Chocolate Fudge	0,65	230
7211	Master Origins Colombia	0,69	230
7216	Altissio	0,48	40
7218	Diavolitto	0,48	40
7219	Dolce	0,55	80
7225	Scuro	0,55	80
7229	Barista Creations Bianco Doppio	0,56	80
7234	Melozio Decaf	0,64	230
7244	Altissio Decaf	0,50	40

```
      ⍞JSON csv fn_products
DOMAIN ERROR: JSON export: the right argument cannot be converted
      ⍞JSON csv fn_products
      ^
      ]box off
Was ON
```

Exercise 6

Convert the “Products” table to JSON

JSON can’t describe a matrix!

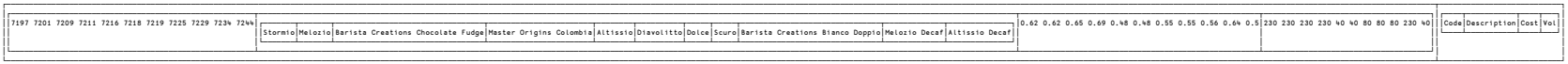
```
]JSON csv products
```

- Explore which of the `]CSV` import formats can be converted

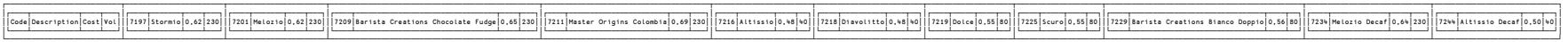
- Explore ways of transforming the matrix

Exercise 6 walk-through

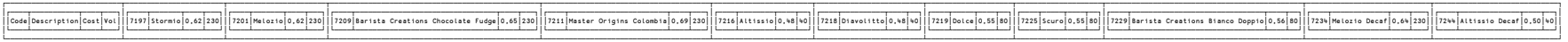
```
      ]box on -style=min
Was OFF -style=min
      ⎕JSON csv fn_products
DOMAIN ERROR: JSON export: the right argument cannot be converted
      ⎕JSON csv fn_products
      ^
      (csv⍠'Invert' 2) fn_products '' (2 1 2 2) 1
```



```
      ⎕JSON (csv⍠'Invert' 2) fn_products '' (2 1 2 2) 1
[[[7197,7201,7209,7211,7216,7218,7219,7225,7229,7234,7244],["Stormio","Melozio","Barista Creations Chocolate Fudge","Master Origins
Colombia","Altissio","Diavolitto","Dolce","Scuro","Barista Creations Bianco Doppio","Melozio Decaf","Altissio
Decaf"],[0.62,0.62,0.65,0.69,0.48,0.48,0.55,0.55,0.56,0.64,0.5],[230,230,230,230,40,40,80,80,80,230,40]],["Code","Description","Cost","Vol"]]
      ↓csv fn_products
```



```
      ⎕JSON ↓csv fn_products
[["Code","Description","Cost","Vol"],["7197","Stormio","0,62","230"],["7201","Melozio","0,62","230"],["7209","Barista Creations Chocolate
Fudge","0,65","230"],["7211","Master Origins
Colombia","0,69","230"],["7216","Altissio","0,48","40"],["7218","Diavolitto","0,48","40"],["7219","Dolce","0,55","80"],["7225","Scuro","0,55","80
"],["7229","Barista Creations Bianco Doppio","0,56","80"],["7234","Melozio Decaf","0,64","230"],["7244","Altissio Decaf","0,50","40"]]
      ⎕JSON ⎕JSON ↓csv fn_products
```



```
      (⎕JSON⍠'HighRank' 'Split')csv fn_products
[["Code","Description","Cost","Vol"],["7197","Stormio","0,62","230"],["7201","Melozio","0,62","230"],["7209","Barista Creations Chocolate
Fudge","0,65","230"],["7211","Master Origins
Colombia","0,69","230"],["7216","Altissio","0,48","40"],["7218","Diavolitto","0,48","40"],["7219","Dolce","0,55","80"],["7225","Scuro","0,55","80
"],["7229","Barista Creations Bianco Doppio","0,56","80"],["7234","Melozio Decaf","0,64","230"],["7244","Altissio Decaf","0,50","40"]]
      ]box off
Was ON
```

JSON

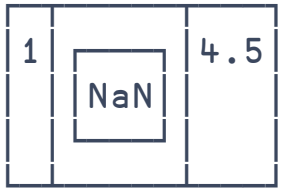
- Text containing structured data:
  - Numbers
  - Strings (character arrays)
  - Objects (namespaces)
  - Vectors
  - Booleans, null
  - Infinity, NaN

JSON types without APL equivalents

```
      json5 'NaN'
```

“Wrapper”

```
      json5 '[1, NaN, 4.5]'
```

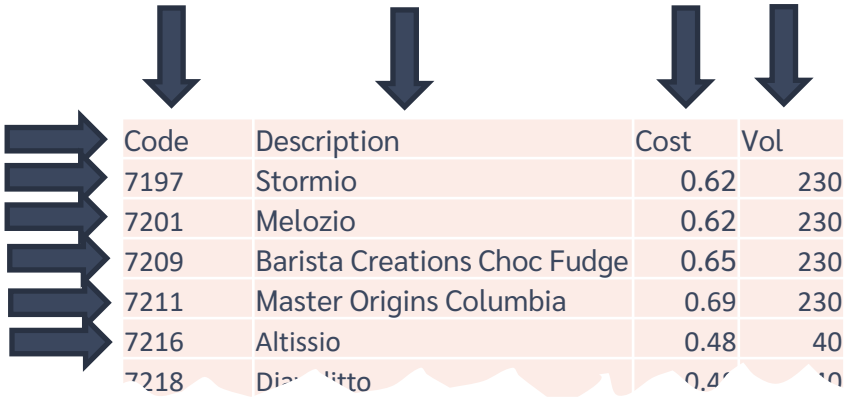



```
      1 json5 1 (⊂'NaN') 4.5
[1,NaN,4.5]
      1 json5 1 (⊂1 'NaN') 4.5
[1,NaN,4.5]
```

Explicitly
“Wrapper type 1”

Another form for JSON tables

- Specifically, tables with:
  - Some columns
  - One header row
  - Some rows of data



Code	Description	Cost	Vol
7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Choc Fudge	0.65	230
7211	Master Origins Columbia	0.69	230
7216	Altissio	0.48	40
7218	Di… …itto	0.4…	…0

Another form for JSON tables

A table can be represented as:

- An array of objects
- Where each object contains named items
- And each item’s name is a column name (header value)
- And each value is the cell content

For example…

Code	Description	Cost	Vol
7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Choc Fudge	0.65	230
7211	Master Origins Columbia	0.69	230
7216	Altissio	0.48	40
7218	Di… …itto	0.4…	…0


```
[
  {
    "Code": 7197,
    "Description": "Stormio",
    "Cost": 0.62,
    "Vol": 230
  },
  {
    "Code": 7201,
    "Description": "Melozio",
    "Cost": 0.62,
    "Vol": 230
  },
...
  {
    "Code": 7244,
    "Description": "Altissio Decaf",
    "Cost": 0.5,
    "Vol": 40
  }
]
```


```
<Products>
  <Product>
    <Code>  7197  </Code>
    <Description>   Stormio   </Description>
    <Cost>  0.62  </Cost>
    <Vol>  230  </Vol>
  </Product>
  <Product>
    <Code>  7201  </Code>
    <Description>   Melozio   </Description>
    <Cost>  0.62  </Cost>
    <Vol>  230  </Vol>
  </Product>
...
  <Product>
    <Code>  7244  </Code>
    <Description>   Altissio Decaf   </Description>
    <Cost>  0.5  </Cost>
    <Vol>  40  </Vol>
  </Product>
</Products>
```

```
[
  {
    "Code": 7197,
    "Description": "Stormio",
    "Cost": 0.62,
    "Vol": 230
  },
  {
    "Code": 7201,
    "Description": "Melozio",
    "Cost": 0.62,
    "Vol": 230
  },
...
  {
    "Code": 7244,
    "Description": "Altissio Decaf",
    "Cost": 0.5,
    "Vol": 40
  }
]
```

Demo 7

```
      ]box on -style=min
Was OFF -style=min
      ns←(⎕NS'')(⎕NS'')(⎕NS'')
      ns[1].(Code Description Cost Vol)←7197 'Stormio' 0.62 630
      ns[2].(Code Description Cost Vol)←7201 'Melozio' 0.62 230
      ns[3].(Code Description Cost Vol)←7244 'Altissio Decaf' 0.5 40
      (⎕JSON⍠'Compact' 0) ns
[
  {
    "Code": 7197,
    "Cost": 0.62,
    "Description": "Stormio",
    "Vol": 630
  },
  {
    "Code": 7201,
    "Cost": 0.62,
    "Description": "Melozio",
    "Vol": 230
  },
  {
    "Code": 7244,
    "Cost": 0.5,
    "Description": "Altissio Decaf",
    "Vol": 40
  }
]
      ⍝ ? create namespace array from matrix format ?
      ⎕VR 'build_ns'
     ∇ ret←build_ns m;rows;cols;r;c
[1]    rows cols←⍴m
[2]    ret←{⎕NS''}¨⍳rows-1
[3]    :For r :In ⍳rows-1
[4]        :For c :In ⍳cols
[5]            ⍎'ret[',(⍕r),'].',(⊃d[1;c]),'←''',(⍕⊃d[r+1;c]),''''
[6]        :EndFor
```

```
[7]    :EndFor
      ∇
       ⎕←d←csv fn_products '' 4
```

Code	Description	Cost	Vol
7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Chocolate Fudge	0.65	230
7211	Master Origins Colombia	0.69	230
7216	Altissio	0.48	40
7218	Diavolitto	0.48	40
7219	Dolce	0.55	80
7225	Scuro	0.55	80
7229	Barista Creations Bianco Doppio	0.56	80
7234	Melozio Decaf	0.64	230
7244	Altissio Decaf	0.5	40

```
       build_ns d
 #.[Namespace]  #.[Namespace]  #.[Namespace]  #.[Namespace]  #.[Namespace]  #.[Namespace]  #.[Namespace]
#.[Namespace]  #.[Namespace]  #.[Namespace]  #.[Namespace]
       ⎕JSON build_ns d
```


```
[{"Code":"7197","Cost":"0.62","Description":"Stormio","Vol":"230"},{"Code":"7201","Cost":"0.62","Description":"Melozio","Vol":"230"},{"Code":"720
9","Cost":"0.65","Description":"Barista Creations Chocolate Fudge","Vol":"230"},{"Code":"7211","Cost":"0.69","Description":"Master Origins
Colombia","Vol":"230"},{"Code":"7216","Cost":"0.48","Description":"Altissio","Vol":"40"},{"Code":"7218","Cost":"0.48","Description":"Diavolitto",
"Vol":"40"},{"Code":"7219","Cost":"0.55","Description":"Dolce","Vol":"80"},{"Code":"7225","Cost":"0.55","Description":"Scuro","Vol":"80"},{"Code"
:"7229","Cost":"0.56","Description":"Barista Creations Bianco Doppio","Vol":"80"},{"Code":"7234","Cost":"0.64","Description":"Melozio
```

Decaf","Vol":"230"},{"Code":"7244","Cost":"0.5","Description":"Altissio Decaf","Vol":"40"}]

```
      ⎕JSON ⊂2 d
```

[{"Code":7197,"Description":"Stormio","Cost":0.62,"Vol":230},{"Code":7201,"Description":"Melozio","Cost":0.62,"Vol":230},{"Code":7209,"Descriptio
n":"Barista Creations Chocolate Fudge","Cost":0.65,"Vol":230},{"Code":7211,"Description":"Master Origins
Colombia","Cost":0.69,"Vol":230},{"Code":7216,"Description":"Altissio","Cost":0.48,"Vol":40},{"Code":7218,"Description":"Diavolitto","Cost":0.48,
"Vol":40},{"Code":7219,"Description":"Dolce","Cost":0.55,"Vol":80},{"Code":7225,"Description":"Scuro","Cost":0.55,"Vol":80},{"Code":7229,"Descrip
tion":"Barista Creations Bianco Doppio","Cost":0.56,"Vol":80},{"Code":7234,"Description":"Melozio
Decaf","Cost":0.64,"Vol":230},{"Code":7244,"Description":"Altissio Decaf","Cost":0.5,"Vol":40}]

```
      ⎕←d←csv fn_products '' (2 1 2 2) 1
```

7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Chocolate Fudge	0.65	230
7211	Master Origins Colombia	0.69	230
7216	Altissio	0.48	40
7218	Diavolitto	0.48	40
7219	Dolce	0.55	80
7225	Scuro	0.55	80
7229	Barista Creations Bianco Doppio	0.56	80
7234	Melozio Decaf	0.64	230
7244	Altissio Decaf	0.5	40

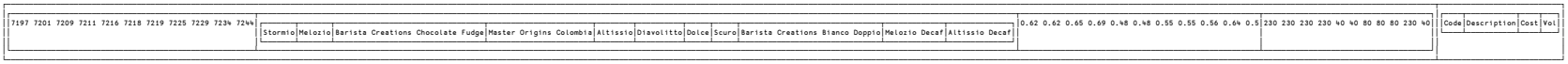
Code	Description	Cost	Vol

```
      ⎕JSON ⊂3 d
```

[{"Code":7197,"Description":"Stormio","Cost":0.62,"Vol":230},{"Code":7201,"Description":"Melozio","Cost":0.62,"Vol":230},{"Code":7209,"Descriptio
n":"Barista Creations Chocolate Fudge","Cost":0.65,"Vol":230},{"Code":7211,"Description":"Master Origins
Colombia","Cost":0.69,"Vol":230},{"Code":7216,"Description":"Altissio","Cost":0.48,"Vol":40},{"Code":7218,"Description":"Diavolitto","Cost":0.48,

"Vol":40},{"Code":7219,"Description":"Dolce","Cost":0.55,"Vol":80},{"Code":7225,"Description":"Scuro","Cost":0.55,"Vol":80},{"Code":7229,"Description":"Barista Creations Bianco Doppio","Cost":0.56,"Vol":80},{"Code":7234,"Description":"Melozio Decaf","Cost":0.64,"Vol":230},{"Code":7244,"Description":"Altissio Decaf","Cost":0.5,"Vol":40}]

```
      ⍞←d←(csv⍠'Invert' 2) fn_products '' (2 1 2 2) 1
```




```
      ⍞JSON ⊂4 d
```

[{"Code":7197,"Description":"Stormio","Cost":0.62,"Vol":230},{"Code":7201,"Description":"Melozio","Cost":0.62,"Vol":230},{"Code":7209,"Description":"Barista Creations Chocolate Fudge","Cost":0.65,"Vol":230},{"Code":7211,"Description":"Master Origins Colombia","Cost":0.69,"Vol":230},{"Code":7216,"Description":"Altissio","Cost":0.48,"Vol":40},{"Code":7218,"Description":"Diavolitto","Cost":0.48,"Vol":40},{"Code":7219,"Description":"Dolce","Cost":0.55,"Vol":80},{"Code":7225,"Description":"Scuro","Cost":0.55,"Vol":80},{"Code":7229,"Description":"Barista Creations Bianco Doppio","Cost":0.56,"Vol":80},{"Code":7234,"Description":"Melozio Decaf","Cost":0.64,"Vol":230},{"Code":7244,"Description":"Altissio Decaf","Cost":0.5,"Vol":40}]

```
      ]box off
Was ON
```

Agenda

- Text files
- CSV files
- Portable file functions
- JSON
- **XML**

⍯XML

- Read and write XML text.
- Does not read and write files.

Code	Description	Cost	Vol
7197	Stormio	0.62	230
7201	Melozio	0.62	230
7209	Barista Creations Choc Fudge	0.65	230
7211	Master Origins Columbia	0.69	230
7216	Altissio	0.48	40
7218	Di… …tto	0.4…	…0

```
<Products>
  <Product>
    7197
    <Description> Stormio </Description>
    <Cost> 0.62 </Cost>
    230
  </Product>
  <Product>
    <Code> 7201 </Code>
    <Hatstand> Melozio </Hatstand>
    <Cost> 0.62 </Cost>
    <Vol> 230 </Vol>
  </Product>
...
  <Product>
    <Code> 7244 </Code>
    <Description> Altissio Decaf </Description>
    <Cost Currency="GBP"> 0.50 </Cost>
    <Vol> 40 </Vol>
  </Product>
</Products>
```

XML

- Convert to an easier-to-process matrix.
  - Describes the XML itself!

```
0  Products                      3
1  Product                       3
2  Code          7197            5
2  Description   Stormio         5
2  Cost          0.62            5
2  Vol           230             5

1  Product                       3
2  Code          7201            5
2  Description   Melozio         5
2  Cost          0.62            5
2  Vol           230             5

...
1  Product                       3
2  Code          7244            5
2  Description   Altissio Decaf  5
2  Cost          0.5             5
2  Vol           40              5
```

```
<Products>
  <Product>
    <Code> 7197 </Code>
    <Description> Stormio </Description>
    <Cost> 0.62 </Cost>
    <Vol> 230 </Vol>
  </Product>
  <Product>
    <Code> 7201 </Code>
    <Description> Melozio </Description>
    <Cost> 0.62 </Cost>
    <Vol> 230 </Vol>
  </Product>
...
  <Product>
    <Code> 7244 </Code>
    <Description> Altissio Decaf </Description>
    <Cost> 0.5 </Cost>
    <Vol> 40 </Vol>
  </Product>
</Products>
```

```
0  Products                    3
1  Product                     3
2  Code          7197          5
2  Description   Stormio       5
2  Cost          0.62          5
2  Vol           230           5

1  Product                     3
2  Code          7201          5
2  Description   Melozio       5
2  Cost          0.62          5
2  Vol           230           5

...
1  Product                     3
2  Code          7244          5
2  Description   Altissio Decaf 5
2  Cost          0.5           5
2  Vol           40            5
```

Five-column matrix:

- Level
- Element name
- Character data (if any)
- Attributes
- Row type

Row type:

3 – element with sub-elements

5 – element with character data

Exercise 7

Import FILE8.txt (products) & FILE9.txt (orders)

Then:

Experiment with processing the data, e.g.:

- List the individual products.
- List the orders, with product names.

```
⎕XML ⊃⎕NGET 'FILE8.txt'
```

Exercise 7 walk-through

```
      products←⎕XML ⍕⎕NGET 'FILE8.txt'
      orders←⎕XML ⍕⎕NGET 'FILE9.txt'
      )ed products
      ⎕←m←(⊂'Description')≡¨products[;2]
0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0
      ⎕←i←m/⍳⍴products
4 9 14 19 24 29 34 39 44 49 54
      ↑products[i;3]
Stormio
Melozio
Barista Creations Chocolate Fudge
Master Origins Colombia
Altissio
Diavolitto
Dolce
Scuro
Barista Creations Bianco Doppio
Melozio Decaf
Altissio Decaf
      ⍝ List the orders, with product names
      products[i-1;3]
 7197  7201  7209  7211  7216  7218  7219  7225  7229  7234  7244
      {(products[i-1;3])≡¨⊂⍵}'7209'
0 0 1 0 0 0 0 0 0 0 0
      {⍕((products[i-1;3])≡¨⊂⍵)/products[i;3]}'7209'
Barista Creations Chocolate Fudge
      name←{⍕((products[i-1;3])≡¨⊂⍵)/products[i;3]}
      )ed orders
      ⎕←m←(⊂'Date')≡¨orders[;2]
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
      ⎕←j←m/⍳⍴orders
3 7 11 15 19 23 27 31 35 39 43 47 51 55 59 63 67 71 75 79 83 87 91 95 99 103
      ↑{(⍕(orders[⍵;3])),': ',(⍕(orders[⍵+2;3])),' of ',(⍕(orders[⍵+1;3]))}¨j
12/09/2023: 50 of 7216
12/09/2023: 30 of 7218
```

```
07/08/2023: 50 of 7225
21/06/2023: 30 of 7197
21/06/2023: 10 of 7201
21/06/2023: 10 of 7234
21/06/2023: 50 of 7225
21/06/2023: 50 of 7216
21/06/2023: 50 of 7211
19/06/2023: 30 of 7197
19/06/2023: 20 of 7218
19/06/2023: 50 of 7219
19/06/2023: 20 of 7229
12/04/2023: 50 of 7234
12/04/2023: 20 of 7201
12/04/2023: 30 of 7244
12/04/2023: 40 of 7219
12/04/2023: 20 of 7218
12/04/2023: 40 of 7229
23/11/2023: 10 of 7209
23/11/2023: 10 of 7197
23/11/2023: 50 of 7201
23/11/2023: 50 of 7234
23/11/2023: 40 of 7244
23/11/2023: 10 of 7218
23/11/2023: 30 of 7219
      †{(⊃(orders[ω;3])),': ',(⊃(orders[ω+2;3])),' of ',name(⊃(orders[ω+1;3]))}¨j
12/09/2023: 50 of Altissio
12/09/2023: 30 of Diavolitto
07/08/2023: 50 of Scuro
21/06/2023: 30 of Stormio
21/06/2023: 10 of Melozio
21/06/2023: 10 of Melozio Decaf
21/06/2023: 50 of Scuro
21/06/2023: 50 of Altissio
21/06/2023: 50 of Master Origins Colombia
19/06/2023: 30 of Stormio
19/06/2023: 20 of Diavolitto
```

```
19/06/2023: 50 of Dolce
19/06/2023: 20 of Barista Creations Bianco Doppio
12/04/2023: 50 of Melozio Decaf
12/04/2023: 20 of Melozio
12/04/2023: 30 of Altissio Decaf
12/04/2023: 40 of Dolce
12/04/2023: 20 of Diavolitto
12/04/2023: 40 of Barista Creations Bianco Doppio
23/11/2023: 10 of Barista Creations Chocolate Fudge
23/11/2023: 10 of Stormio
23/11/2023: 50 of Melozio
23/11/2023: 50 of Melozio Decaf
23/11/2023: 40 of Altissio Decaf
23/11/2023: 10 of Diavolitto
23/11/2023: 30 of Dolce
      ⍝ About that matrix ...
      ]box off
Was OFF
      ⎕XML ⍠⎕NGET 'FILE8.txt'
0  Products                                          3
1  Product                                           3
2  Code          7197                                5
2  Description   Stormio                             5
2  Cost          0.62                                5
2  Vol           230                                 5
1  Product                                           3
2  Code          7201                                5
2  Description   Melozio                             5
2  Cost          0.62                                5
2  Vol           230                                 5
1  Product                                           3
2  Code          7209                                5
2  Description   Barista Creations Chocolate Fudge   5
2  Cost          0.65                                5
2  Vol           230                                 5
1  Product                                           3
```

```
2  Code         7211                              5
2  Description  Master Origins Colombia           5
2  Cost         0.69                              5
2  Vol          230                               5
1  Product                                        3
2  Code         7216                              5
2  Description  Altissio                          5
2  Cost         0.48                              5
2  Vol          40                                5
1  Product                                        3
2  Code         7218                              5
2  Description  Diavolitto                        5
2  Cost         0.48                              5
2  Vol          40                                5
1  Product                                        3
2  Code         7219                              5
2  Description  Dolce                            5
2  Cost         0.55                              5
2  Vol          80                                5
1  Product                                        3
2  Code         7225                              5
2  Description  Scuro                            5
2  Cost         0.55                              5
2  Vol          80                                5
1  Product                                        3
2  Code         7229                              5
2  Description  Barista Creations Bianco Doppio   5
2  Cost         0.56                              5
2  Vol          80                                5
1  Product                                        3
2  Code         7234                              5
2  Description  Melozio Decaf                     5
2  Cost         0.64                              5
2  Vol          230                               5
1  Product                                        3
2  Code         7244                              5
```

```
2  Description  Altissio Decaf                          5
2  Cost         0.5                                    5
2  Vol          40                                     5
      j
[
  {
    "Code": 7197,
    "Description": "Stormio",
    "Cost": 0.62,
    "Vol": 230
  },
  {
    "Code": 7201,
    "Description": "Melozio",
    "Cost": 0.62,
    "Vol": 230
  },
  {
    "Code": 7209,
    "Description": "Barista Creations Chocolate Fudge",
    "Cost": 0.65,
    "Vol": 230
  },
  {
    "Code": 7211,
    "Description": "Master Origins Colombia",
    "Cost": 0.69,
    "Vol": 230
  },
  {
    "Code": 7216,
    "Description": "Altissio",
    "Cost": 0.48,
    "Vol": 40
  },
  {
```

```
    "Code": 7218,
    "Description": "Diavolitto",
    "Cost": 0.48,
    "Vol": 40
  },
  {
    "Code": 7219,
    "Description": "Dolce",
    "Cost": 0.55,
    "Vol": 80
  },
  {
    "Code": 7225,
    "Description": "Scuro",
    "Cost": 0.55,
    "Vol": 80
  },
  {
    "Code": 7229,
    "Description": "Barista Creations Bianco Doppio",
    "Cost": 0.56,
    "Vol": 80
  },
  {
    "Code": 7234,
    "Description": "Melozio Decaf",
    "Cost": 0.64,
    "Vol": 230
  },
  {
    "Code": 7244,
    "Description": "Altissio Decaf",
    "Cost": 0.5,
    "Vol": 40
  }
]
```

```
      ⎕JSON j
 #.[JSON object]  #.[JSON object]  #.[JSON object]  #.[JSON object]  #.[JSON object]  #.[JSON object]  #.[JSON
object]  #.[JSON object]  #.[JSON object]  #.[JSON object]  #.[JSON object]
      (⎕JSON⍠'Format' 'M') j
0                                                    2
1                                                    1
2  Code                                   7197       3
2  Description                            Stormio    4
2  Cost                                      0.62    3
2  Vol                                     230       3
1                                                    1
2  Code                                   7201       3
2  Description                            Melozio    4
2  Cost                                      0.62    3
2  Vol                                     230       3
1                                                    1
2  Code                                   7209       3
2  Description  Barista Creations Chocolate Fudge    4
2  Cost                                      0.65    3
2  Vol                                     230       3
1                                                    1
2  Code                                   7211       3
2  Description            Master Origins Colombia    4
2  Cost                                      0.69    3
2  Vol                                     230       3
1                                                    1
2  Code                                   7216       3
2  Description                           Altissio    4
2  Cost                                      0.48    3
2  Vol                                      40       3
1                                                    1
2  Code                                   7218       3
2  Description                          Diavolitto    4
2  Cost                                      0.48    3
2  Vol                                      40       3
1                                                    1
```

2	Code	7219	3
2	Description	Dolce	4
2	Cost	0.55	3
2	Vol	80	3
1			1
2	Code	7225	3
2	Description	Scuro	4
2	Cost	0.55	3
2	Vol	80	3
1			1
2	Code	7229	3
2	Description	Barista Creations Bianco Doppio	4
2	Cost	0.56	3
2	Vol	80	3
1			1
2	Code	7234	3
2	Description	Melozio Decaf	4
2	Cost	0.64	3
2	Vol	230	3
1			1
2	Code	7244	3
2	Description	Altissio Decaf	4
2	Cost	0.5	3
2	Vol	40	3

Please give feedback!

http://is.gd/D23workshop

Password: **D23**

Full address:
https://questionpro.com/t/AYklwZzukX