DYALOG

APL Germany

Array Notation RC1

Adám Brudzewsky

Die Saga geht weiter…

DYALOG

APL Germany

Array Notation RC1

Adám Brudzewsky

Warum?

- Vermeidung komplizierter Ausdrücke, um Felder zu erzeugen
  Passen oft nicht in eine Zeile

Warum jetzt?

- Vermeidung komplizierter Ausdrücke, um Felder zu erzeugen
  Passen oft nicht in eine Zeile
- Definition von Feldern von Quellcode-Management
  Änderungen werden oft zeilenbasiert behandelt

Warum jetzt?

- Vermeidung komplizierter Ausdrücke, um Felder zu erzeugen
  Passen oft nicht in eine Zeile
- Definition von Feldern von Quellcode-Management
  Änderungen werden oft zeilenbasiert behandelt
- Felder in Textform
  Bearbeiten mit beliebigem Editor, Mailclients etc.

Was?

- Mittelgroße Felder
  Leere und triviale Felder besser mit Ausdrücken handhaben

Was?

- Mittelgroße Felder
  Leere und triviale Felder besser mit Ausdrücken handhaben

- Felder mit höherem Rang
  Wir haben gute Notation für Vektoren und kleine Vektoren von Vektoren

Was?

- Mittelgroße Felder
  Leere und triviale Felder besser mit Ausdrücken handhaben
- Felder mit höherem Rang
  Wir haben gute Notation für Vektoren und kleine Vektoren von Vektoren
- Tiefer als 2

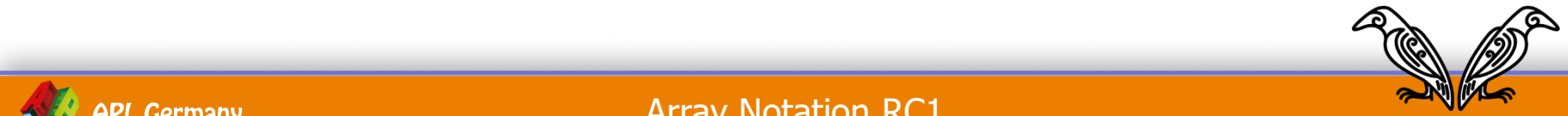
RC1⁉


2015
2016
2017
2018
2020
Phil Last
Sizilien
Dyalog Mk I
intern
Dyalog Mk II
Helsingör
Dyalog Mk III
Belfast
Acre-3 Mk I
Acre-4 Mk I
Acre-4 Mk II
RC1

RC1⁉

2020

RC1



RC1?

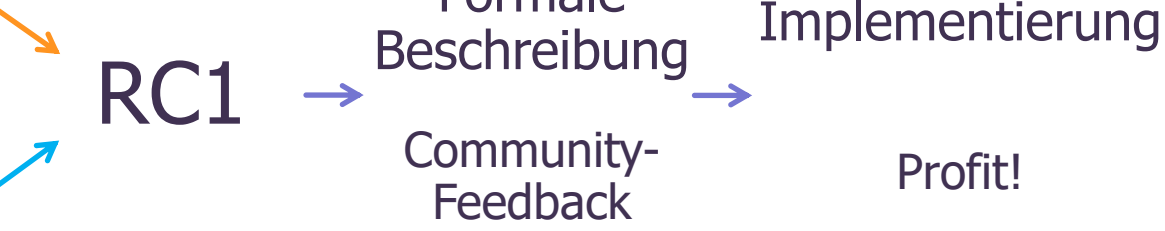
2020 2021

RC1 → Formale Beschreibung

Community-Feedback

RC1⁉

2020 2021 2022



Wo?

Wo?

- Link
  - Teil Ihrer Dyalog-Installation

Wo?

- Link
  Teil Ihrer Dyalog-Installation
- Acre
  github.com/the-carlisle-group/Acre-Desktop

Acre

Acre

```
      ⎕←var←⍳2 3
```

1	1	1	2	1	3
2	1	2	2	2	3

Acre

```
      ⎕←var←⍳2 3
```

1	1	1	2	1	3
2	1	2	2	2	3

```
      ]CreateProject C:\tmp\acretest #
#
```

Acre

```
      ⎕←var←⍳2 3
```

1	1	1	2	1	3
2	1	2	2	2	3

```
      ]CreateProject C:\tmp\acretest #
#
      ]SetChanged var
```

#.var

Acre

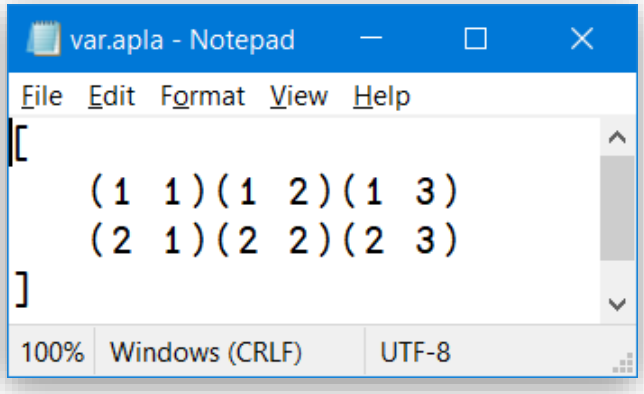
```
      ⎕←var←⍳2 3
```

1	1	1	2	1	3
2	1	2	2	2	3

```
      ]CreateProject C:\tmp\acretest #
#
      ]SetChanged var
```

#.var

```
      ]Open "C:\tmp\acretest\APLSource\var.apla" -using=notepad
```



Acre

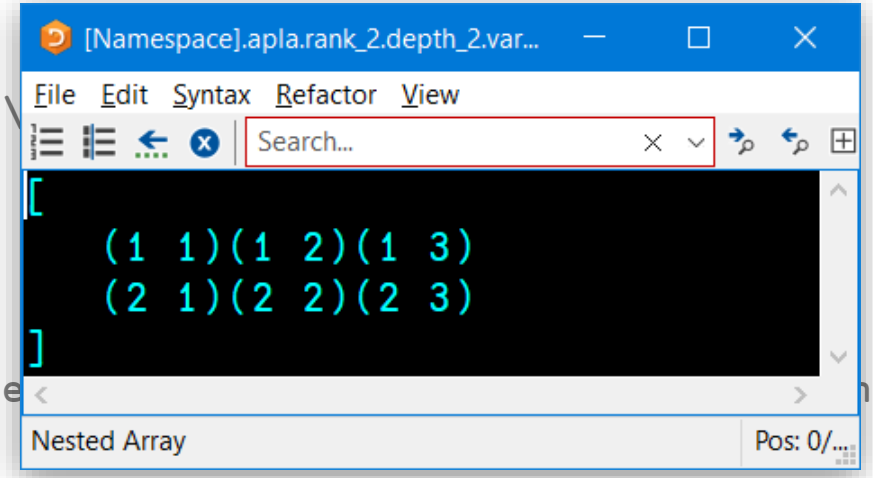
```
      ⎕←var←⍳2 3
```

1	1	1	2	1	3
2	1	2	2	2	3

```
      ]CreateProject C:\
#
      ]SetChanged var
```

#.var

```
      ]Open "C:\tmp\acre                    notepad
      ]EditArray var
```



Link

Link

```
      ]Link.Create # C:\tmp\acretest\APLSource
Linked: # ↔ C:\tmp\acretest\APLSource
```

Link

```
      ]Link.Create # C:\tmp\acretest\APLSource
Linked: # ↔ C:\tmp\acretest\APLSource
      var
```

1	1	1	2	1	3
2	1	2	2	2	3

Link

```
      ]Link.Create # C:\tmp\acretest\APLSource
Linked: # ↔ C:\tmp\acretest\APLSource
      var
```

1 1	1 2	1 3
2 1	2 2	2 3

```
      ⎕SE.Link.Serialise var
[(1 1)  (1 2)  (1 3)
 (2 1)  (2 2)  (2 3)]
```

Link

Link

```
      ↑ s←'([1 2 ◇ 3 4]' ' [5 6 ◇ 7 8])'
([1 2 ◇ 3 4]
 [5 6 ◇ 7 8])
```

Link

```
      ⍝ s←'([1 2 ⋄ 3 4]' ' [5 6 ⋄ 7 8])'
([1 2 ⋄ 3 4]
 [5 6 ⋄ 7 8])

      ⎕SE.Link.Deserialise s
```

1 2	5 6
3 4	7 8

Link

```
      ⍝ s←'([1 2 ⋄ 3 4]' ' [5 6 ⋄ 7 8])'
([1 2 ⋄ 3 4]
 [5 6 ⋄ 7 8])

      ⎕SE.Link.Deserialise s
```

1 2	5 6
3 4	7 8

```
      {['ABC' ⋄ 'DEF']}⎕SE.Link.Array⍬
ABC
DEF
```

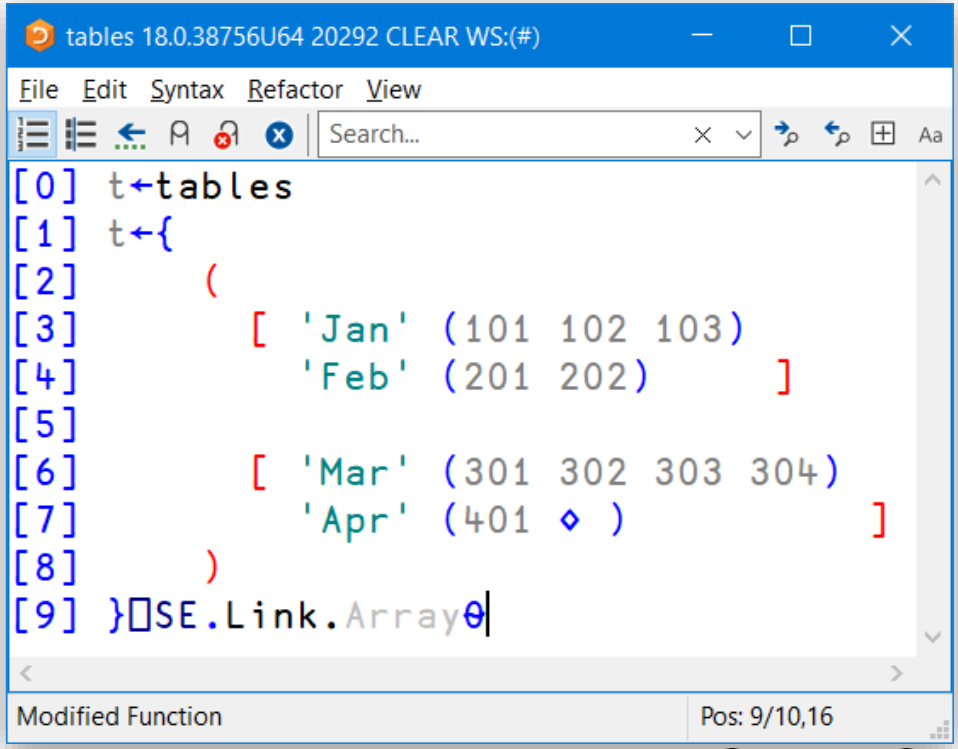
Link

Link

```
)ed tables
```

Link

```
)ed tables
```

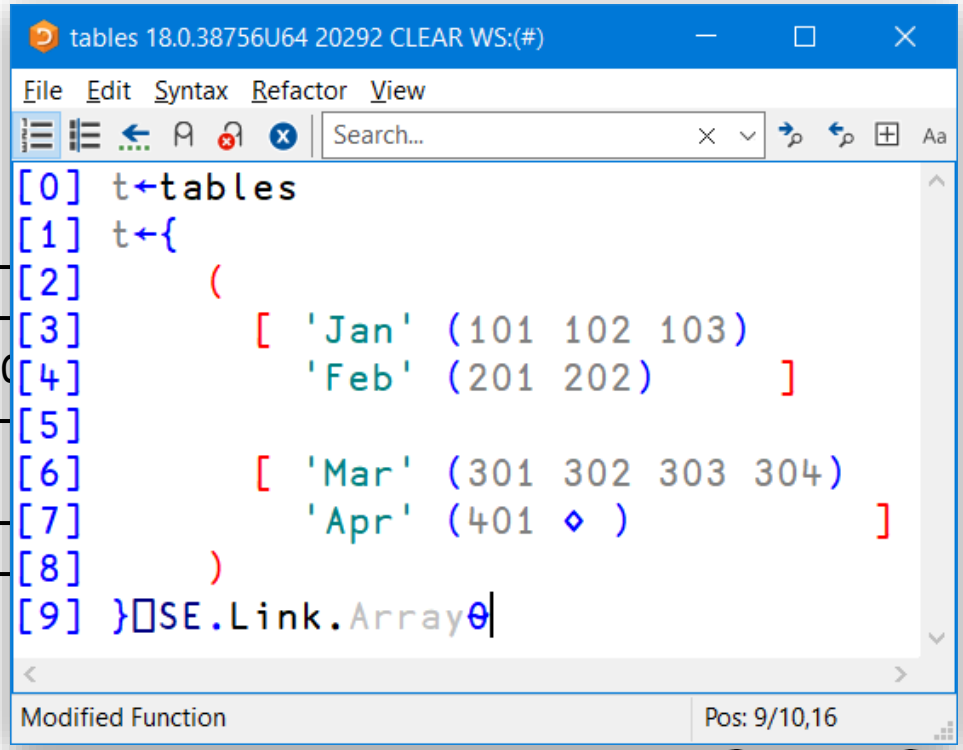


Link

```
)ed tables
tables
```

Jan	101 102 103
Feb	201 202

Mar	301 302 30
Apr	401



Tutorial

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck
- Neu: runde Klammern ohne Ausdrücke

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck
- Neu: runde Klammern ohne Ausdrücke
- Ausdrücke getrennt durch ⋄ oder Zeilenvorschub

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck
- Neu: runde Klammern ohne Ausdrücke
- Ausdrücke getrennt durch ⋄ oder Zeilenvorschub
- Runde Klammern: `( a ⋄ b ⋄ c )`

  Jeder nicht-leere Ausdruck erzeugt ein Element im neuen Vektor

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck
- Neu: runde Klammern ohne Ausdrücke
- Ausdrücke getrennt durch ⋄ oder Zeilenvorschub
- Runde Klammern: `( a ⋄ b ⋄ c )`
  Jeder nicht-leere Ausdruck erzeugt ein Element im neuen Vektor
- Eckige Klammern: `[ a ⋄ b ⋄ c ]`
  Jeder nicht-leere Ausdruck erzeugt eine „major cell” (Rang ≥ 1) im neuen Feld

Tutorial

- Neu: runde und eckige Klammern enthalten mehr als 1 Ausdruck
- Neu: runde Klammern ohne Ausdrücke
- Ausdrücke getrennt durch ⋄ oder Zeilenvorschub
- Runde Klammern: `( a ⋄ b ⋄ c )`
  Jeder nicht-leere Ausdruck erzeugt ein Element im neuen Vektor
- Eckige Klammern: `[ a ⋄ b ⋄ c ]`
  Jeder nicht-leere Ausdruck erzeugt eine „major cell" (Rang ≥ 1) im neuen Feld
- Runde Klammern, die leer sind oder mindestens ein `:` enthalten
  Jeder Ausdruck ist einen Namen/Werte-Paar, getrennt durch `:`

dfns.dws: cal

```
Q1←'January' 'February' 'March    '~¨' '     ⍝ 1st quarter month names.
Q2←'April  ' 'May     ' 'June     '~¨' '     ⍝ 2nd   ..       ..     ..
Q3←'July   ' 'August  ' 'September'~¨' '     ⍝ 3rd   ..       ..     ..
Q4←'October' 'November' 'December '~¨' '     ⍝ 4th   ..       ..     ..
months←Q1,Q2,Q3,Q4                           ⍝ month names for year.
```

dfns.dws: cal

```
months←(                                    ⍝ month names for year.
    'January'◇'February'◇'March'            ⍝ 1st quarter month names.
    'April'  ◇'May'     ◇'June'             ⍝ 2nd   ..      ..    ..
    'July'   ◇'August'  ◇'September'        ⍝ 3rd   ..      ..    ..
    'October'◇'November'◇'December'          ⍝ 4th   ..      ..    ..
)
```

dfns.dws: morse

```
{⍵~¨' '}⍵\¨{          ⍝ plain-text and Morse codes.
 ('A' '.-   ')('B' '-... ')('C' '-.-. ')('D' '-..  '),⍵}{
 ('E' '.    ')('F' '..-. ')('G' '--.  ')('H' '.... '),⍵}{
 ('I' '..   ')('J' '.--- ')('K' '-.-  ')('L' '.-.. '),⍵}{
 ('M' '--   ')('N' '-.   ')('O' '---  ')('P' '.--. '),⍵}{
 ('Q' '--.- ')('R' '.-.  ')('S' '...  ')('T' '-    '),⍵}{
 ('U' '..-  ')('V' '...- ')('W' '.--  ')('X' '-..- '),⍵}{
 ('Y' '-.-- ')('Z' '--.. '),⍵}{

 ('0' '-----')('1' '.---- ')('2' '..--- ')('3' '...-- '),⍵}{
 ('4' '....-')('5' '..... ')('6' '-.... ')('7' '--... '),⍵}{
 ('8' '---..')('9' '----. '),⍵}{

 ('.' '.-.-.- ')(',' '--..-- ')(':' '---... '),⍵}{
 ('?' '..--.. ')('''' '.----. ')('-' '-....- '),⍵}{
 ('/' '-..-.  ')('(' '-.--.  ')(')' '-.--.- '),⍵}{
 ('"' '.-..-. ')('@' '.--.-. ')('=' '-...- '),⍵}{

 ⍵}⊂' ' ' / '          ⍝ a blank / inter-word separator.
```

dfns.dws: morse

```
(                      ⍝ plain-text and Morse codes.
  'A'  '.-'    ⋄ 'B'  '-...' ⋄ 'C'  '-.-.' ⋄ 'D'  '-..'
  'E'  '.'     ⋄ 'F'  '..-.' ⋄ 'G'  '--.'  ⋄ 'H'  '....'
  'I'  '..'    ⋄ 'J'  '.---' ⋄ 'K'  '-.-'  ⋄ 'L'  '.-..'
  'M'  '--'    ⋄ 'N'  '-.'   ⋄ 'O'  '---'  ⋄ 'P'  '.--.'
  'Q'  '--.-'  ⋄ 'R'  '.-.'  ⋄ 'S'  '...'  ⋄ 'T'  '-'
  'U'  '..-'   ⋄ 'V'  '...-' ⋄ 'W'  '.--'  ⋄ 'X'  '-..-'
  'Y'  '-.--'  ⋄ 'Z'  '--..'

  '0'  '-----' ⋄ '1'  '.----' ⋄ '2'  '..---' ⋄ '3'  '...--'
  '4'  '....-' ⋄ '5'  '.....' ⋄ '6'  '-....' ⋄ '7'  '--...'
  '8'  '---..' ⋄ '9'  '----.'

  '.'  '.-.-.-' ⋄ ','  '--..--' ⋄ ':'  '---...'
  '?'  '..--..' ⋄ ''''  '.----.' ⋄ '-'  '-....-'
  '/'  '-..-.'  ⋄ '('  '-.--.'  ⋄ ')'  '-.--.-'
  '"'  '.-..-.' ⋄ '@'  '.--.-.' ⋄ '='  '-...-'

  ' '  ' / '  )        ⍝ a blank / inter-word separator.
```

dfns.dws: morse

```
(                      ⍝ plain-text and Morse codes.
  'A'  '.-'    ◇ 'B'  '-...' ◇ 'C'  '-.-.' ◇ 'D'  '-..'
  'E'  '.'     ◇ 'F'  '..-.' ◇ 'G'  '--.'  ◇ 'H'  '....'
  'I'  '..'    ◇ 'J'  '.---' ◇ 'K'  '-.-'  ◇ 'L'  '.-..'
  'M'  '--'    ◇ 'N'  '-.'   ◇ 'O'  '---'  ◇ 'P'  '.--.'
  'Q'  '--.-'  ◇ 'R'  '.-.'  ◇ 'S'  '...'  ◇ 'T'  '-'
  'U'  '..-'   ◇ 'V'  '...-' ◇ 'W'  '.--'  ◇ 'X'  '-..-'
  'Y'  '-.--'  ◇ 'Z'  '--..'

  '0'  '-----' ◇ '1'  '.----' ◇ '2'  '..---' ◇ '3'  '...--'
  '4'  '....-' ◇ '5'  '.....' ◇ '6'  '-....' ◇ '7'  '--...'
  '8'  '---..' ◇ '9'  '----.'

  '.'  '.-.-.-' ◇ ','  '--..--' ◇ ':'  '---...'
  '?'  '..--..' ◇ ''''  '.----.' ◇ '-'  '-....-'
  '/'  '-..-.'  ◇ '('  '-.--.'  ◇ ')'  '-.--.-'
  '"'  '.-..-.' ◇ '@'  '.--.-.' ◇ '='  '-...-'

  ' '  ' / '  )        ⍝ a blank / inter-word separator.
```

dfns.dws: morse

```
(                      ⍝ plain-text and Morse codes.
  'A'  '.-'    ◇ 'B'  '-...' ◇ 'C'  '-.-.' ◇ 'D'  '-..' ↵
  'E'  '.'     ◇ 'F'  '..-.' ◇ 'G'  '--.'  ◇ 'H'  '....'
  'I'  '..'    ◇ 'J'  '.---' ◇ 'K'  '-.-'  ◇ 'L'  '.-..'
  'M'  '--'    ◇ 'N'  '-.'   ◇ 'O'  '---'  ◇ 'P'  '.--.'
  'Q'  '--.-'  ◇ 'R'  '.-.'  ◇ 'S'  '...'  ◇ 'T'  '-'
  'U'  '..-'   ◇ 'V'  '...-' ◇ 'W'  '.--'  ◇ 'X'  '-..-'
  'Y'  '-.--'  ◇ 'Z'  '--..'

  '0'  '-----' ◇ '1'  '.----' ◇ '2'  '..---' ◇ '3'  '...--'
  '4'  '....-' ◇ '5'  '.....' ◇ '6'  '-....' ◇ '7'  '--...'
  '8'  '---..' ◇ '9'  '----.'

  '.'  '.-.-.-' ◇ ','  '--..--' ◇ ':'  '---...'
  '?'  '..--..' ◇ ''''  '.----.' ◇ '-'  '-....-'
  '/'  '-..-.'  ◇ '('  '-.--.'  ◇ ')'  '-.--.-'
  '"'  '.-..-.' ◇ '@'  '.--.-.' ◇ '='  '-...-'

  ' '  ' / '  )        ⍝ a blank / inter-word separator.
```

dfns.dws: morse

```
(                      ⍝ plain-text and Morse codes.
  'A'  '.-'    ◇ 'B'  '-...' ◇ 'C'  '-.-.' ◇ 'D'  '-..'
  'E'  '.'     ◇ 'F'  '..-.' ◇ 'G'  '--.'  ◇ 'H'  '....'
  'I'  '..'    ◇ 'J'  '.---' ◇ 'K'  '-.-'  ◇ 'L'  '.-..'
  'M'  '--'    ◇ 'N'  '-.'   ◇ 'O'  '---'  ◇ 'P'  '.--.'
  'Q'  '--.-'  ◇ 'R'  '.-.'  ◇ 'S'  '...'  ◇ 'T'  '-'
  'U'  '..-'   ◇ 'V'  '...-' ◇ 'W'  '.--'  ◇ 'X'  '-..-'
  'Y'  '-.--'  ◇ 'Z'  '--..' ↵
↵
  '0'  '-----' ◇ '1'  '.----' ◇ '2'  '..---' ◇ '3'  '...--'
  '4'  '....-' ◇ '5'  '.....' ◇ '6'  '-....' ◇ '7'  '--...'
  '8'  '---..' ◇ '9'  '----.'

  '.'  '.-.-.-' ◇ ','  '--..--' ◇ ':'  '---...'
  '?'  '..--..' ◇ ''' '.----.' ◇ '-'  '-....-'
  '/'  '-..-.'  ◇ '('  '-.--.'  ◇ ')'  '-.--.-'
  '"'  '.-..-.' ◇ '@'  '.--.-.' ◇ '='  '-...-'

  ' '  ' / '  )        ⍝ a blank / inter-word separator.
```

math.dws: Eigen

```
⍵⍵{⍺,⊂' <C1    ' 'V'}{          ⍝ JOBZ
   ⍺,⊂' <C1    ' 'L'}{          ⍝ UPLO
   ⍺,⊂' <I4    'n}{             ⍝ N
   ⍺,⊂' =F8[;] '(⍴⍺⍵mat)}{      ⍝ A
   ⍺,⊂' <I4    'n}{             ⍝ LDA
   ⍺,⊂' >F8[] 'n}{              ⍝ W
   ⍺,⊂' >F8[] '(¯2+4×n)}{       ⍝ WORK
   ⍺,⊂' <I4    '(¯1+2×n)}{      ⍝ LWORK
   ⍺,⊂' >F8[] '(¯2+3×n)}{       ⍝ RWORK
   ⍺,⊂' >I4    '0}⍬             ⍝ INFO
```

math.dws: Eigen

```
[ ' <C1 ' 'V'                    ⍝ JOBZ
  ' <C1 ' 'L'                    ⍝ UPLO
  ' <I4 '  n                     ⍝ N
  ' =F8[] '(⍎⍕mat)               ⍝ A
  ' <I4 '  n                     ⍝ LDA
  ' >F8[] 'n                     ⍝ W
  ' >F8[] '(¯2+4×n)              ⍝ WORK
  ' <I4 '  (¯1+2×n)              ⍝ LWORK
  ' >F8[] '(¯2+3×n)              ⍝ RWORK
  ' >I4 '  0                     ⍝ INFO
]
```

Profile ucmd: DBMenuCB

```
poss←1 2⍴'fns'((0 1)(0.7 0)(0.7 0)×size)
poss,←'fnd'((0 1)(0 0)(0 0)×size)
poss,←'lines'((0 0)(0.7 0)(0.7 0)×size)
poss,←'lnd'((0 0)(0 0)(0 0)×size)
```

Profile ucmd: DBMenuCB

```
poss←['fns'   ((0.0 1 ◇ 0.7 0 ◇ 0.7 0)×size)
      'fnd'   ((0.0 1 ◇ 0.0 0 ◇ 0.0 0)×size)
      'lines'((0.0 0 ◇ 0.7 0 ◇ 0.7 0)×size)
      'lnd'   ((0.0 0 ◇ 0.0 0 ◇ 0.0 0)×size)]
```

Profile ucmd: DBMenuCB

```
poss←['fns'   ((0.0 1 ◇ 0.7 0 ◇ 0.7 0)×size)
      'fnd'   ((0.0 1 ◇ 0.0 0 ◇ 0.0 0)×size)
      'lines'((0.0 0 ◇ 0.7 0 ◇ 0.7 0)×size)
      'lnd'   ((0.0 0 ◇ 0.0 0 ◇ 0.0 0)×size)]
```

```
[3 2◇4 1][[2 1◇2 2];[1 2◇1 1]]
```

Profile ucmd: DBMenuCB

```
poss←['fns'   ((0.0 1 ◇ 0.7 0 ◇ 0.7 0)×size)
      'fnd'   ((0.0 1 ◇ 0.0 0 ◇ 0.0 0)×size)
      'lines'((0.0 0 ◇ 0.7 0 ◇ 0.7 0)×size)
      'lnd'   ((0.0 0 ◇ 0.0 0 ◇ 0.0 0)×size)]
```

```
[3 2◇4 1][[2 1◇2 2];[1 2◇1 1]]
```

SALT: SettingsTable

```
UserFolder←'[HOME]',(WIN↓'/'),'MyUCMDs'
CmdDir←UserFolder,PATHDEL[1],SALTFOLDER,FS,'spice'
split←{1↓¨(s∈ω)⊂s←(ω∈1↑α)↓ω,α}
:field shared SettingsTable←0 5⍴''
⍝ name; description; registry name; default; value
⍝ e.g. [ProgramFiles]BeyondCompare
SettingsTable⍪←'compare;the comparison program to use;CompareCMD;APL;'split';'
⍝ Cmd Folders are locations where Spice commands are stored - their existence is not challenged
⍝ 2nd folder is something like this: C:\Users\DanB2\Documents\Dyalog APL 14.0 Unicode Files
SettingsTable⍪←('cmddir⊗list of Spice folders (commands) to use separated by ',PATHDEL[1],'⊗CommandFolder⊗',CmdDir,'⊗')split'⊗'
SettingsTable⍪←'debug;debug level;DebugLevel;0;0' split';'
SettingsTable⍪←'editor;the editor program to use;EditorCMD;notepad;'split';'
SettingsTable⍪←'edprompt;whether the editor prompts for confirmation;EdPrompt;1;' split';'
SettingsTable⍪←'fndels;whether tradfns are saved enclosed in ∇s;FnDels;0;'split';'
SettingsTable⍪←'mapprimitives;whether to map some primitives to ⎕Uxxxx on Classic;MapPrim;1;' split';'
⍝ automatic for all but PI
SettingsTable⍪←('newcmd;detection of new user commands;CmdDetect;',(⊃Pi⍵'auto' 'manual'),';') split';'
SettingsTable⍪←'track;saving of new items and which info to stored in SALT tags;Track;;' split';'
⍝ only APL and XML so far
SettingsTable⍪←'varfmt;whether variables are saved as XML docs or APL expressions;VarFmt;xml;' split';'
⍝ WorkFolders are locations where files are searched - their existence is not challenged
SettingsTable⍪←('workdir⊗list of storage directories to use (separated by ',PATHDEL[1],')⊗SourceFolder⊗',SALTFOLDER,'⊗')split'⊗'
```

SALT: SettingsTable

```
UserFolder←'[HOME]',(WIN↓'/'),'MyUCMDs'
CmdDir←UserFolder,PATHDEL[1],SALTFOLDER,FS,'spice'

:field shared SettingsTable←[
⍝ name          ⍝ description                                                    ⍝ registry name ⍝ default              ⍝ value
⍝ e.g. [ProgramFiles]BeyondCompare
'compare'       'the comparison program to use'                                  'CompareCMD'    'APL'                ''
⍝ Cmd Folders are locations where Spice commands are stored - their existence is not challenged
⍝ 2nd folder is something like this: C:\Users\DanB2\Documents\Dyalog APL 14.0 Unicode Files
'cmddir'        (list of Spice folders (commands) to use separated by ',PATHDEL[1]) 'CommandFolder' CmdDir              ''
'debug'         'debug level'                                                    'DebugLevel'    '0'                  '0'
'editor'        'the editor program to use'                                      'EditorCMD'     'notepad'            ''
'edprompt'      'whether the editor prompts for confirmation'                    'EdPrompt'      '1'                  ''
'fndels'        'whether tradfns are saved enclosed in ∇s'                       'FnDels'        '0'                  ''
'mapprimitives' 'whether to map some primitives to ⎕Uxxxx on Classic'            'MapPrim'       '1'                  ''
⍝ automatic for all but PI
'newcmd'        'detection of new user commands'                                 'CmdDetect'     (⊃Pi⍵'auto' 'manual') ''
'track'         'saving of new items and which info to store in SALT tags'       'Track'         ''                   ''
⍝ only APL and XML so far
'varfmt'        'whether variables are saved as XML docs or APL expressions'     'VarFmt'        'xml'                ''
⍝ WorkFolders are locations where files are searched - their existence is not challenged
'workdir'       ('list of storage directories to use (separated by ',PATHDEL[1],')') 'SourceFolder' SALTFOLDER        ''
]
```

Link: DefaultOpts

```
(
  beforeRead:        ''
  beforeWrite:       ''
  caseCode:          0
  codeExtensions:    ( 'aplf'
                       'aplo'
                       'apln'
                       'aplc'
                       'apli'
                       'dyalog'
                       'apl'
                       'mipage' )
  customExtensions: ''
  flatten:           0
  forceExtensions:   0
  forceFilenames:    0
  source:            'dir'
  typeExtensions:    [ 2   'apla'
                       3   'aplf'
                       4   'aplo'
                       9.1 'apln'
                       9.4 'aplc'
                       9.5 'apli' ]
  watch:             'ns'
)
```

Link: Export

```
(defopts←⎕NS ⍬).(overwrite)←0
```

```
defopts←(overwrite:0)
```

Link: Import

```
opts←⎕NS ⍬
```

```
opts←()
```

500€ Preisgeld zu gewinnen

500€ Preisgeld zu gewinnen

```
      ⎕SE.Link.Serialise ↓3 4⍴⎕A
( 'ABCD'
  'EFGH'
  'IJKL' )
```

```
      ⎕SE.Link.Serialise ⍳¨⍳3
( ( 1 ◇)
  1 2
  1 2 3  )
```

```
      ⎕SE.Link.Serialise ↓2 3 2⍴⎕A
['AB'  'CD'  'EF'
 'GH'  'IJ'  'KL']
```

```
      ⎕SE.Link.Serialise ↓2 3 2⍴⍳10
[(1 2)  (3 4)  (5 6)
 (7 8)  (9 10) (1 2)]
```

Mitmachen und 5 × 100€ gewinnen

Mitmachen und 5 × 100€ gewinnen

```
      ⎕SE.Link.Serialise  ⍪↓⍪⎕TC
[ (( (⎕UCS 8) ⋄ ) ⋄ )
  (( (⎕UCS 10) ⋄ ) ⋄ )
  (( (⎕UCS 13) ⋄ ) ⋄ ) ]
```

```
      ⎕SE.Link.Serialise  ↓↓2 3 2⍴⍳10
( ( 1 2 ⋄ 3 4 ⋄ 5 6 )
  ( 7 8 ⋄ 9 10 ⋄ 1 2  ) )
```

Mitmachen und 5 × 100€ gewinnen

```
   ⎕SE.Link.Serialise  ⍪↓⍪⎕TC
[ (( (⎕UCS 8) ⋄ ) ⋄ )
  (( (⎕UCS 10) ⋄ ) ⋄ )
  (( (⎕UCS 13) ⋄ ) ⋄ ) ]
          ⇩
[((⎕UCS  8 ⋄ ) ⋄ )
 ((⎕UCS 10 ⋄ ) ⋄ )
 ((⎕UCS 13 ⋄ ) ⋄ )]
```

```
   ⎕SE.Link.Serialise  ↓↓2 3 2⍴⍳10
( ( 1 2 ⋄ 3 4 ⋄ 5 6 )
  ( 7 8 ⋄ 9 10 ⋄ 1 2  ) )
          ⇩
((1 2 ⋄ 3  4 ⋄ 5 6)
 (7 8 ⋄ 9 10 ⋄ 1 2))
```

Matrizen & Felder höherer Rang

Mehrzeilig		Inline		Ausdruck

`[1 2`
`3 4`
`5 6]`
	⇔	`[1 2 ⋄ 3 4 ⋄ 5 6]`	⇔	`3 2⍴1 2 3 4 5 6`

`[1`
`2`
`3]`
	⇔	`[1 ⋄ 2 ⋄ 3]`	⇔	`3 1⍴1 2 3`

Vektoren & genesteten Felder

Mehrzeilig		Inline		Ausdruck

`(1 2`
`3 4`
`5 6)`
	⇔	`(1 2 ⋄ 3 4 ⋄ 5 6)`	⇔	`(1 2)(3 4)(5 6)`

`(1`
`2`
`3)`
	⇔	`(1 ⋄ 2 ⋄ 3)`	⇔	`1 2 3`

Namensräume (Namespaces)

Mehrzeilig		Inline		Ausdruck

`(`
`  a:'APL'`
`  b:c,c←1 2`
`)`
	⇔	`(a:'APL' ⋄ b:c,c←1 2)`	⇔	`{`
				`  α←⎕NS⍬`
				`  α.a←'APL'`
				`  α.b←{`
				`  c,c←1 2`
				`  }⍬`
				`  α`
				`}⍬`

`(` `)`	⇔	`()`	⇔	`⎕NS⍬`

Fragen? *Auf Englisch, bitte!*

DYALOG

APL Germany

```
[1 2 ◇ 3 4 ◇ 5 6]
(1 2 ◇ 3 4 ◇ 5 6)
(a:'APL' ◇ b:c,c←1 2)
```

aplwiki.com/wiki/array notation

Adám Brudzewsky
adam@dyalog.com

Array notation

Array notation is a way to write most arrays literally, with no or minimal use of primitive functions, possibly over multiple code lines. While APL has had at least simple numeric Strand notation since APL\360, no major APL has implemented native support for an extended notation as of 2020.

Medium-sized array constants are often needed in code. Due to the lack of a native multi-line notation, programmers have resorted to various ad-hoc methods of approximating such, usually at the cost of reduced readability. A very common technique is repeated concatenation:

Fragen? *Auf Englisch, bitte!*

DYALOG

Fortsetzung folgt!

APL Germany

```
[1 2 ⋄ 3 4 ⋄ 5 6]
(1 2 ⋄ 3 4 ⋄ 5 6)
(a:'APL' ⋄ b:c,c←1 2)
```

aplwiki.com/wiki/array notation
Adám Brudzewsky
adam@dyalog.com