



Mainz, April 30th 2024

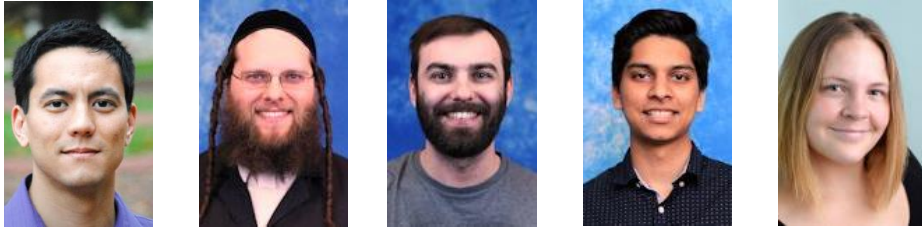


Array Notation

Morten Kromberg, CTO

Dyalog – The Next Generation

2010-2021



2022



2023



2024



2023 Summer Interns

Dyalog – The Next Generation

2010-2021



2022

2023

2024

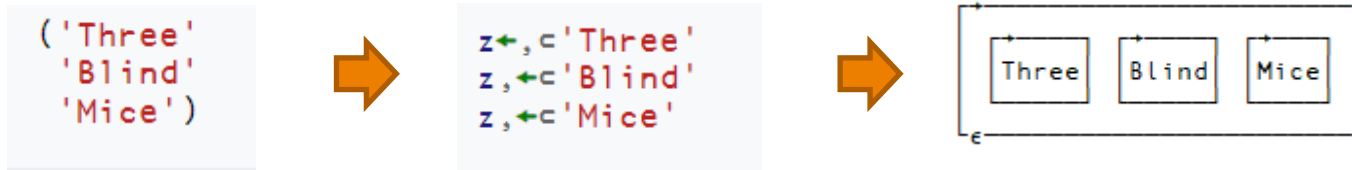


2023 Summer Interns

A Way to Write Most Arrays *Literally*

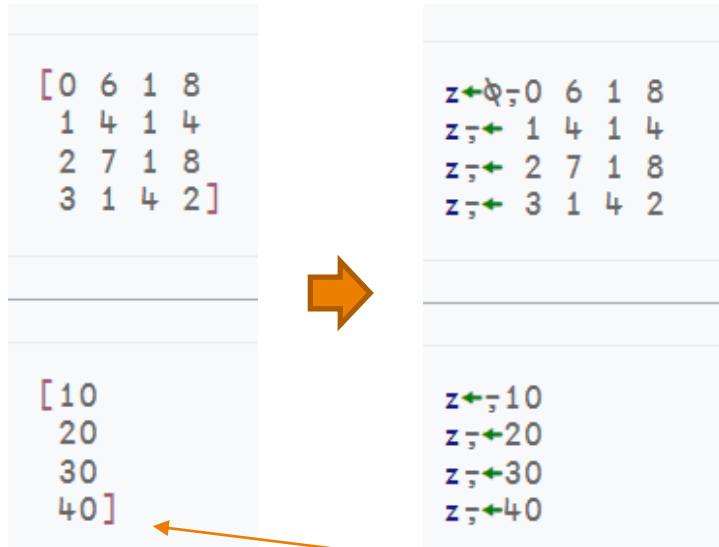
- ◆ *Literally* means: Statically parseable, without executing APL primitives
- ◆ Nested vectors using (...)
- ◆ High rank arrays using [...]
- ◆ Namespaces using (**name:value** pairs)

Hopefully "Intuitive" ... let's test it!



- Multiple statements within parentheses define a nested vector

Matrices



- Statements within brackets define a high rank array
- The results of the statements are "mixed" to produce the final array
- NB: The minimum rank of each individual result is 1 (vector)
 - The result will have 2 or more dimensions.
- The last example gives us a one-column matrix, not a vector

"Multiple Statements"

```
[ 1  0  1
  0  1  0
  1  0  1]
```

... is equivalent to ...

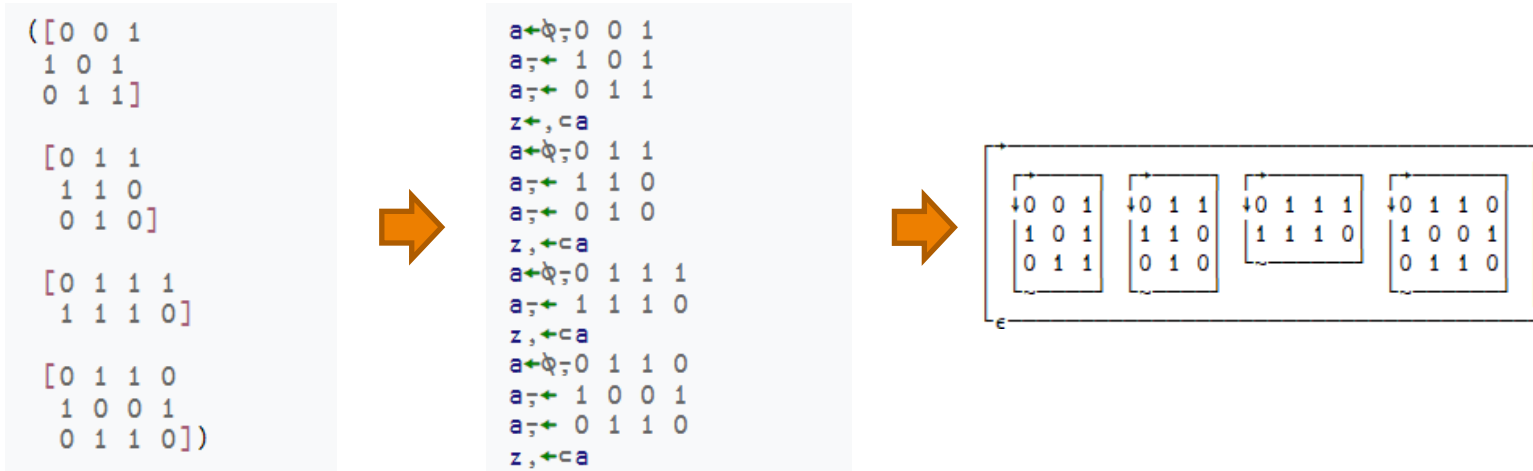
```
[ 1  0  1  ♦  0  1  0  ♦  1  0  1 ]
```

Array Notation uses brackets and parentheses...

... that span Multiple Statements.

Statements can be separated by new lines OR by diamonds

Vector of Matrices



- A vector (...) of four matrices [...]

Error Message Table

```
[0 'OK'  
1 'WS FULL'  
2 'SYNTAX ERROR'  
3 'INDEX ERROR'  
4 'RANK ERROR']
```



```
z←0 0 'OK'  
z←1 1 'WS FULL'  
z←2 2 'SYNTAX ERROR'  
z←3 3 'INDEX ERROR'  
z←4 4 'RANK ERROR'
```

- A two-column nested matrix

Literal Namespace Notation

```
person+(first: 'Max' ♦ last: 'Mustermann')  
person.last, ', ', person.first  
Mustermann, Max
```

```
person+(  
  first: 'Max'  
  last: 'Mustermann'  
)
```

🟡 "JSON for APL"

Namespace With a Matrix

```
(y:(x:['hello'  
      'world']))
```



```
z←⊝NS⊖  
z.y←⊝NS⊖  
z.y.x←⊖; 'hello'  
z.y.x;← 'world'
```

- Vector notation, Matrix notation, and Namespace notation can be contained within each other.

Empty Namespace(s)

```
    ()  
#. [Namespace]  
    () () ()  
#. [Namespace] #. [Namespace] #. [Namespace]
```

EXPLORER

- JSWC
 - JSWC
 - classes
 - menuitem
 - poly
 - rect
 - ribbon
 - ribbonbutton
 - ribbonbuttongroup
 - ribbongroup
 - ribbongroupitem
 - scroll
 - splitter
 - ClassName.apla
 - Defaults.apla**
 - Dynamic.apla
 - methodlist.apla
 - PropList.apla
 - Supported.apla
 - OUTLINE
 - TIMELINE

PropList.apla

```

1 (
2   'Type'
3   'SplitObj1'
4   'SplitObj2'
5   'Posn'
6   'Size'
7   'Style'
8   'Coord'
9   'Align'
10  'Active'
11  'Visible'
12  'Event'
13  'BCol'
14  'CursorObj'
15  'Data'
16  'KeepOnClose'
17  'MethodList'
18  'ChildList'
19  'EventList'
20  'PropList'
21 )
22

```

Supported.apla

```

1 (
2   'Type'
3   'SplitObj1'
4   'SplitObj2'
5   'Style'
6   'Posn'
7   'Visible'
8   'Event'
9 )
10

```

Defaults.apla

```

1 (
2   'Splitter'
3   ''
4   ''
5   0 500
6   800 3
7   'Vert'
8   'Inherit'
9   'None'
10  1
11  1
12  0 0
13  0
14  ''
15  0
16  0
17  (
18  .. 'Detach'
19  )
20  (
21  .. 'Timer'
22  )

```

C:\devt\jswc\ - Dyalog APL/W-64

File Edit View Window Session Log Action Options Tools Threads Help

WS Object Tool Edit Session APL385 Unicode

Language Bar

```

#.JSWC.Doc.Details'Splitter'
The JSWC implementation of Splitter has some degree of support for:

Event      SplitObj1   Style
*Posn      SplitObj2   Visible

* indicates that the property can change after it has been set.

Supported events:

EndSplit

```

Debugger

Ready... Ins

CurObj: Splitter (Undefined) 8:2 □DQ:0 □TRAP □SI:0 □IO:1 □ML:1

```

Supported.apla x
JSWC > classes > splitter > Supported.apla
You, 4 months ago | 1 a
1  (
2  'Type'
3  'SplitObj1'
4  'SplitObj2'
5  'Style'
6  'Posn'
7  'Visible'
8  'Event'
9  )
10

```

Array Notation - Status

- APL model used by Link to store variables for a couple of years
- `SE.Dyalog.Array.Serialise`
`|Deserialise`
- Public proposal published
 - https://aplwiki.com/wiki/Array_notation
- Very little feedback from the community
- Will be integrated with interpreter in v20.0

Prototypes



V20.0 Production



Live Demos..?