The Rank Operator

Richard Park

The Problem

```
      1 10 100 × 1 2 3   ⋄   1 10 100 × 4 5 6
1 20 300
4 50 600
```

The Problem

```
      1 10 100 × 1 2 3    ⋄    1 10 100 × 4 5 6
1 20 300
4 50 600
      1 10 100 × 2 3⍴⍳6
```

The Problem

```
      1 10 100 × 1 2 3   ⋄   1 10 100 × 4 5 6
1 20 300
4 50 600
      1 10 100 × 2 3⍴⍳6
RANK ERROR: Mismatched left and right argument ranks
      1 10 100×2 3⍴⍳6
              ∧
```

The Problem

```
      1 10 100 × 1 2 3   ⋄   1 10 100 × 4 5 6
1 20 300
4 50 600
      1 10 100 (× on vectors) 2 3⍴⍳6
1 20 300
4 50 600
```

The Problem

```
      1 10 100 × 1 2 3  ⋄  1 10 100 × 4 5 6
1 20 300
4 50 600
      1 10 100 (×⍤1) 2 3⍴⍳6
1 20 300
4 50 600
```

Modern APL

Modern APL

Modern APL

- **leading-axis theory**

⍋⍒⌿⍀⍪⊖≢↑↓⍳∪⎕

Modern APL

- **leading-axis theory** ⍋⍒⌿⍀⍪⊖≢↑↓⍳∪⎕
- dfns

Modern APL

- **leading-axis theory** ⍋⍒⌿⍀⍪⊖≢↑↓⍳∪⎕
- dfns
- trains youtu.be/Enlh5qwwDuY

Cells & Rank

aplwiki.com/wiki/cell

```
      1       ⌷2  3  4⍴⎕A
ABCD
EFGH
IJKL
```

```
      2  3  4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

A subarray which is formed by selecting a ***single index*** *along some number of* ***leading axes*** *and the whole of each* ***trailing axis****.*

Cells & Rank

aplwiki.com/wiki/cell

```
      1 2 ⌷2 3 4⍴⎕A
EFGH
```

```
      2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

*A subarray which is formed by selecting a **single index** along some number of **leading axes** and the whole of each **trailing axis**.*

Cells & Rank

aplwiki.com/wiki/cell

```
      1 2 3⌷2 3 4⍴⎕A
G
```

```
      2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

A subarray which is formed by selecting a ***single index*** *along some number of* ***leading axes*** *and the whole of each* ***trailing axis****.*

Cells vs Axis Model

Dyalog Webinars: Selecting from Arrays

dyalog.tv/Webinar/?v=AgYDvSF2FfU

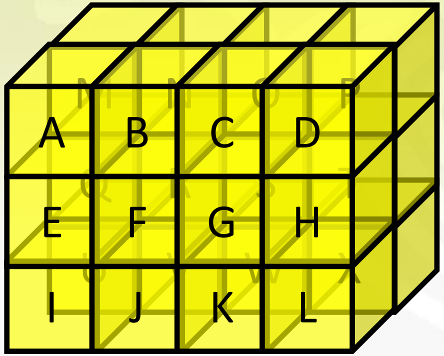
12 minutes, 40 seconds

Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

3D Array

Major cell: Matrix

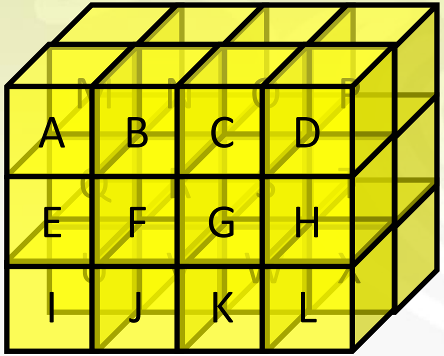


Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

3D Array

Major cell: Matrix

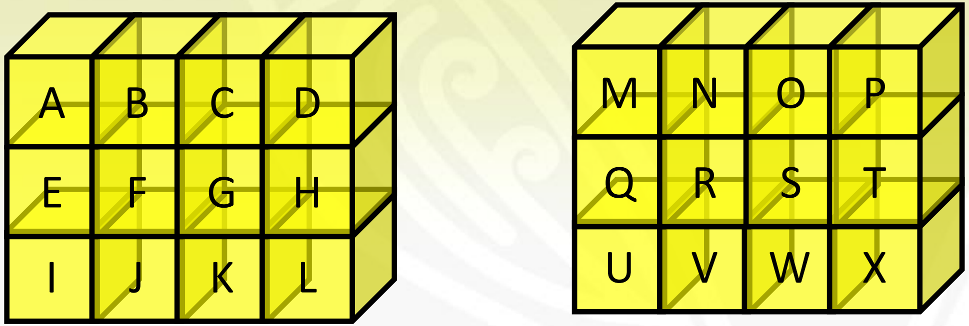


Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

3D Array

Major cell: Matrix

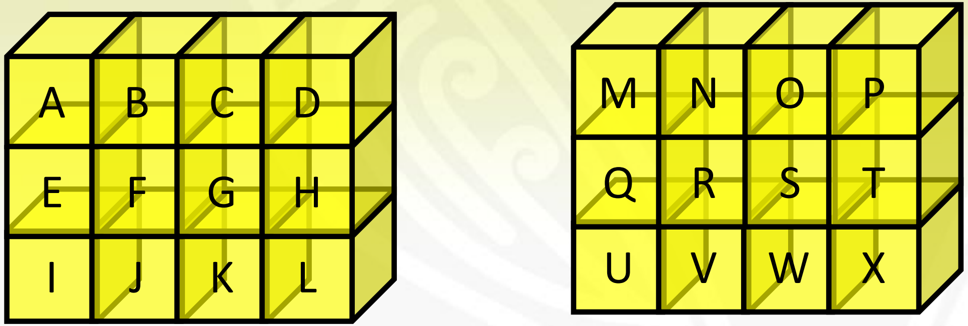


Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

3D Array

Major cell: Matrix



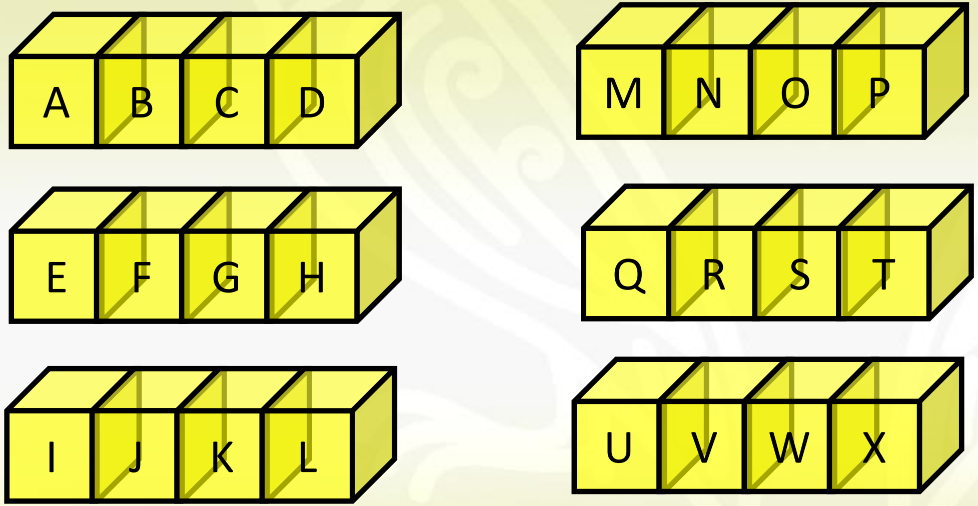
Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

3D Array

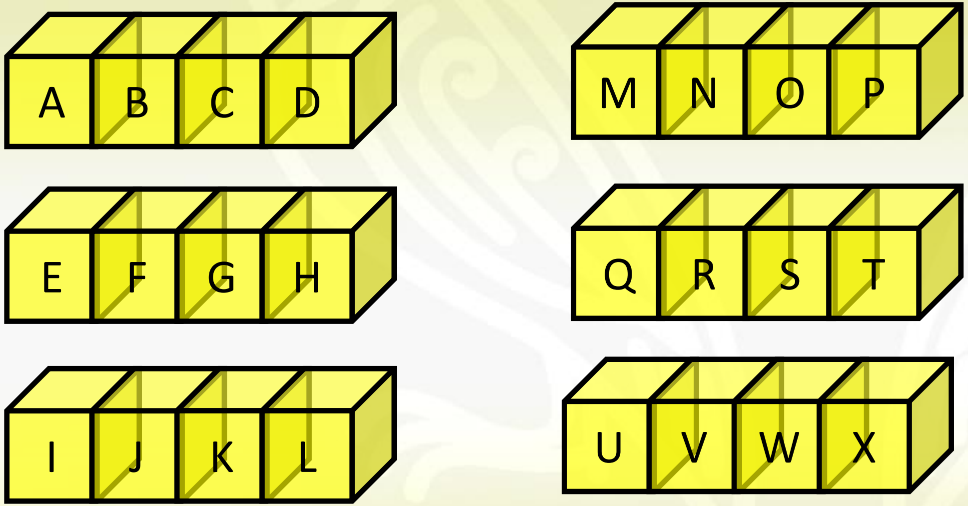
Major cell: Matrix

1-cell: Vector



Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```



3D Array

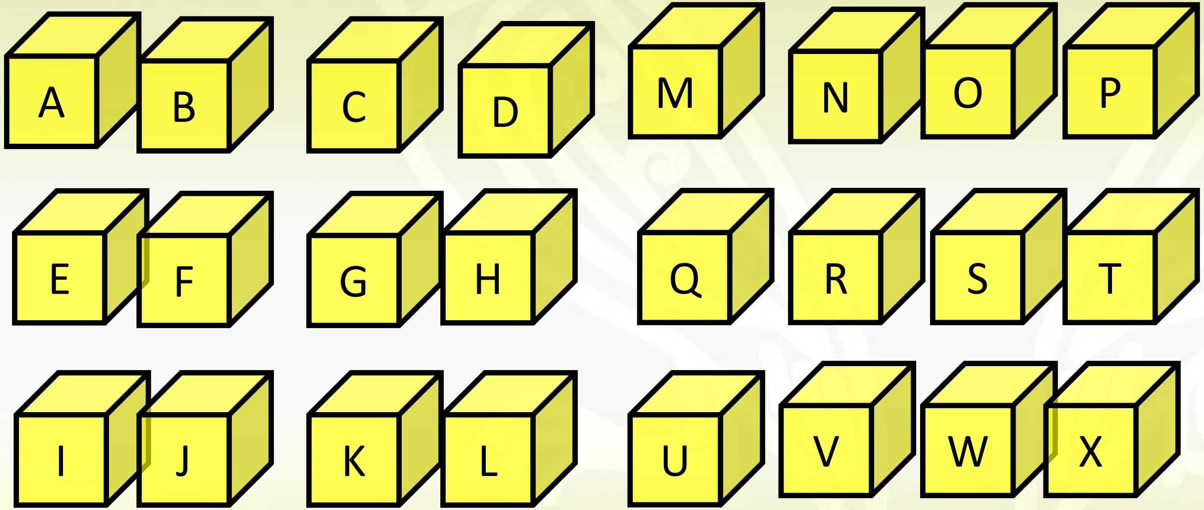
Major cell: Matrix

1-cell: Vector

0-cell: Scalar

Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```



3D Array

Major cell: Matrix

1-cell: Vector

0-cell: Scalar

Hui, R.K. and Kromberg, M.J., 2020

APL Since 1978

Proceedings of the ACM on Programming Languages

Kenneth E. Iverson, 1978

Operators and Functions

Research Report Number #RC7091

Conformable Arrays

```
≡⍤⍴ ∨ 0∊,⍤(≢⍴)
```

Conformable Arrays

```
≡⍤ρ ∨ 1∊,⍤(×/ρ)
```

Uniform Functions f

```
n←2 3⍴0
m←4 5⍴0
```

g⍤⍴ ≡ ⍴⍤f

Uniform Functions f

```
      n←2 3⍴0
      m←4 5⍴0
      ⍴n∘.×m
2 3 4 5
```

```
g⍥⍴ ≡ ⍴⍥f
,⍥⍴ ≡ ⍴⍥(∘.×)
```

Uniform Functions f

```
      n←2 3⍴0
      m←4 5⍴0
      ⍴n∘.×m
2 3 4 5
```

```
g⍤⍴  ≡  ⍴⍤f
,⍤⍴  ≡  ⍴⍤(∘.×)
```

Uniform Functions f

```
      n←2 3⍴0
      m←4 5⍴0
      ⍴n∘.×m
2 3 4 5

      ⍴,n
6
```

```
   g⍤⍴  ≡  ⍴⍤f
   ,⍤⍴  ≡  ⍴⍤(∘.×)
(,×/)⍤⍴  ≡  ⍴⍤,
```

Uniform Functions f

```
      n←2 3⍴0
      m←4 5⍴0
      ⍴n∘.×m
2 3 4 5

      ⍴,n
6

      ⍴⍉n
3 2
```

```
   g⍤⍴  ≡  ⍴⍤f
   ,⍤⍴  ≡  ⍴⍤(∘.×)
(,×/)⍤⍴  ≡  ⍴⍤,
 ⌽⍤⍴  ≡  ⍴⍤⍉
```

Monadic f⍤k

```
⎕←n←2 3 2⍴⍳12
  1  2
  3  4
  5  6

  7  8
  9 10
 11 12
```

```
      ⊖n
  7  8
  9 10
 11 12

  1  2
  3  4
  5  6
```

Monadic f⍤k

```
⎕←n←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

```
      (⊖⍤2)n
 5  6
 3  4
 1  2

11 12
 9 10
 7  8
```

Monadic f⍤k

```
⎕←n←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

```
      (⊖⍤1)n
 2  1
 4  3
 6  5

 8  7
10  9
12 11
```

Monadic f⍤k

```
      n←2 3 2⍴⍳12
      (⊂⍤1)n
┌───┬────┬─────┐
│1 2│3 4 │5 6  │
├───┼────┼─────┤
│7 8│9 10│11 12│
└───┴────┴─────┘
      (+/⍤1)n
 3  7 11
15 19 23
```

Monadic f⍤k

```
      (+/⍤3)n
 3  7 11
15 19 23
      (+/⍤2)n
 3  7 11
15 19 23
      (+/⍤1)n
 3  7 11
15 19 23
```

Monadic f⍤k

```
      (⌽⍤1)n
 2  1
 4  3
 6  5

 8  7
10  9
12 11
```

Monadic f⍤k

```
      (⌽⍤2)n
 2  1
 4  3
 6  5

 8  7
10  9
12 11
```

Monadic f⍤k

```
      (⌽⍤3)n
 2  1
 4  3
 6  5

 8  7
10  9
12 11
```

Monadic f⍥k

```
{ω[⍋ω]}                {(⊂⍋ω)⌷ω}

{(+/ω)÷ρω}             (+⌿÷≢)
```

Monadic f⍤k

```
{⍵[⍋⍵]}                                  {(⊂⍋⍵)⌷⍵}

{(+/⍵)÷⍴⍵}                               (+⌿÷≢)



 ↓   ↓   ↓   ↓   ↓               +/[1]
 1   2   3   4   5
 6   7   8   9  10
11  12  13  14  15
16  17  18  19  20
```

Benefits of Rank

⎕IO independent

User defined functions

Dyadic f⍤l r

```
      1 10 100 × 1 2 3
1 20 300
```

Dyadic f⍤l r

```
      1 10 100 × 1 2 3
1 20 300

      1 10 100 × 2 3⍴⍳6
```

Dyadic f⍤l r

```
      1 10 100 × 1 2 3
1 20 300

      1 10 100 × 2 3⍴⍳6
RANK ERROR: Mismatched left and
right argument ranks
      1 10 100×2 3⍴⍳6
               ∧
```

Dyadic f⍤l r

```
      1 10 100 ×[2] 2 3⍴⍳6
1 20 300
4 50 600
```

Dyadic f⍤l r

```
      1 10 100 ×[2] 2 3⍴⍳6
1 20 300
4 50 600

      1 10 100 ×[2] 1 2 3
```

Dyadic f⍤l r

```
      1 10 100 ×[2] 2 3⍴⍳6
1 20 300
4 50 600

      1 10 100 ×[2] 1 2 3
RANK ERROR: Invalid axis
      1 10 100×[2]1 2 3
               ∧
```

Dyadic f⍤l r

```
      1 10 100(×⍤1 1)1 2 3
1 20 300
```

Dyadic f⍤l r

```
      1 10 100(×⍤1 1)1 2 3
1 20 300
      1 10 100(×⍤1 1)2 3⍴⍳6
1 20 300
4 50 600
```

Dyadic f⍤l r

```
      1 10 100(×⍤1)1 2 3
1 20 300
      1 10 100(×⍤1)2 3⍴⍳6
1 20 300
4 50 600
```

Dyadic f⍤l r

```
      1 10 100 × 1 2 3
1 20 300
      1 10 100 × 3 2⍴⍳6
```

Dyadic f⍤l r

```
      1 10 100 × 1 2 3
1 20 300
      1 10 100 × 3 2⍴⍳6
RANK ERROR: Mismatched left and
right argument ranks
      1 10 100×3 2⍴⍳6
              ∧
```

Dyadic f⍤l r

```
      1 10 100 ×[1] 3 2⍴⍳6
  1    2
 30   40
500  600
```

Dyadic f⍤ l r

```
      1 10 100 ×[1] 3 2ρι6
  1   2
 30  40
500 600
      1 10 100 ×[1] 1 2 3
```

Dyadic f⍤l r

```
      1 10 100 ×[1] 3 2⍴⍳6
 1   2
 30  40
500 600
      1 10 100 ×[1] 1 2 3
1 20 300
```

Dyadic f⍤l r

```
      1 10 100 ×[1] 3 2⍴⍳6
 1    2
 30   40
500 600
      1 10 100 ×[1] 1 2 3
1 20 300
      1 10 100 ×[1] 1
```

Dyadic f⍤l r

```
      1 10 100 ×[1] 3 2⍴⍳6
 1   2
 30  40
500 600
      1 10 100 ×[1] 1 2 3
1 20 300
      1 10 100 ×[1] 1
LENGTH ERROR: Invalid axis
      1 10 100×[1]1
              ∧
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
      1 10 100(×⍤0 1)⍳3
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
      1 10 100(×⍤0 1)⍳3
  1   2   3
 10  20  30
100 200 300
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
      1 10 100(×⍤0 1)⍳3
  1   2   3
 10  20  30
100 200 300
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
      1 10 100(×⍤0 1)⍳3
  1   2   3
 10  20  30
100 200 300
      1 10 100(×⍤0 1)3
```

Dyadic f⍤l r

```
      1 10 100(×⍤0 1)3 2⍴⍳6
  1   2
 30  40
500 600
      1 10 100(×⍤0 1)⍳3
  1   2   3
 10  20  30
100 200 300
      1 10 100(×⍤0 1)3
3 30 300
```

Dyadic f⍤l r

```
      1 10 100 (,⍤⊂⍤0 1) 3 2⍴⍳6   ⍝ {⍺⍵}
┌───┬───┐
│1  │1 2│
├───┼───┤
│10 │3 4│
├───┼───┤
│100│5 6│
└───┴───┘
      1 10 100(+⍤0 1)3 2⍴⍳6
  2   3
 13  14
105 106
```

Dyadic f⍤l r

```
(scalars vectors matrices)←0 1 2
whole←⌊/⍬     ⍝ 99
```

Dyadic f⍤l r

```
(scalars vectors matrices)←0 1 2
whole←⌊/⍬    ⍝ 99
1 10 100(,⍤⊂⍤scalars vectors)3 2⍴⍳6
┌───┬───┐
│1  │1 2│
├───┼───┤
│10 │3 4│
├───┼───┤
│100│5 6│
└───┴───┘
```

Dyadic f⍤l r

```
(scalars vectors matrices)←0 1 2
whole←⌊/θ     ⍝ 99
1 10 100(,⍤⊂⍤scalars matrices)3 2⍴⍳6
┌───┬───┐
│1  │1 2│
│   │3 4│
│   │5 6│
├───┼───┤
│10 │1 2│
│   │3 4│
│   │5 6│
├───┼───┤
│100│1 2│
│   │3 4│
│   │5 6│
└───┴───┘
```

Dyadic f⍤l r

```
(scalars vectors matrices)←0 1 2
whole←⌊/⍬     ⍝ 99
1 10 100(,⍤⊂⍤scalars whole)3 2⍴⍳6
┌───┬───┐
│1  │1 2│
│   │3 4│
│   │5 6│
├───┼───┤
│10 │1 2│
│   │3 4│
│   │5 6│
├───┼───┤
│100│1 2│
│   │3 4│
│   │5 6│
└───┴───┘
```

Outer Product

```
      1 2 3∘.≡1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
```

Outer Product

```
      1 2 3∘.≡1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
      1 2 3(≡⍤0 99)1 2 3 4
0 0 0
```

Outer Product

```
      1 2 3∘.≡1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
      1 2 3(,⍤⊂⍤0 99)1 2 3 4
┌─┬───────┐
│1│1 2 3 4│
├─┼───────┤
│2│1 2 3 4│
├─┼───────┤
│3│1 2 3 4│
└─┴───────┘
```

Outer Product

```
      1 2 3∘.≡1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
      1 2 3(,⍤⊂¨⍤0 99)1 2 3 4
┌───┬───┬───┬───┐
│1 1│1 2│1 3│1 4│
├───┼───┼───┼───┤
│2 1│2 2│2 3│2 4│
├───┼───┼───┼───┤
│3 1│3 2│3 3│3 4│
└───┴───┴───┴───┘
```

Outer Product

```
      1 2 3∘.≡1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
      1 2 3(≡⍤0 99)1 2 3 4
1 0 0 0
0 1 0 0
0 0 1 0
```

Inner Product

```
      _Dot_←{⍺(⍺⍺⌿⍵⍵⍤¯1)⍤1 99⊢⍵}

      (2 3⍴⍳6) +_Dot_× 3 4⍴⍳12
38 44  50  56
83 98 113 128

      (2 3⍴⍳6) +.× 3 4⍴⍳12
38 44  50  56
83 98 113 128
```

Dyalog '09: The Rank Operator
Roger Hui

https://dyalog.tv/Dyalog09/?v=ui76NE5cMWo

Ambivalent f⍥m l r

```
      ⎕←n←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Ambivalent

f⍥m l r

```
      n←2 3 2⍴⍳12
      (⊖⍥3 1 2)n
 7  8
 9 10
11 12

 1  2
 3  4
 5  6
```

Ambivalent

f⍤m l r

```
      n←2 3 2⍴⍳12
      1 2(⊖⍤3 1 2)n
 3  6
 5  2
 1  4

 9 12
11  8
 7 10
```

The Rank Operator Returns

3rd September 2020

Negative Rank	⍤¯k
Multiple Rank	⍤j⍤k
Rank & Transpose	⍤k⊢a⍉

Next Webinar

6th August 2020 (15:00 UTC)

Adám presents

Language Features of Dyalog version 18.0 in Depth (part 4)