Selecting from Arrays

Richard Park (16/04/2020)

What? Indexing? Pfffft…

Selecting from Arrays: Audience

Selecting from Arrays: Audience

Mainly beginners

Selecting from Arrays: Audience

Mainly beginners

Experienced APLers also

Simple

Choose

Reach

The rank operator ⍤

- Sane
- Select
- Choose
- Cells

Simple Choose Reach

Simple Choose Reach

Indexed assignment

```
A←'YEAST' ⋄ A[3 4]←'E'
```

Simple Choose Reach

Indexed assignment `A←'YEAST' ⋄ A[3 4]←'E'`

Modified assignment `A←1 9 ⋄ A[1]+←5`

Simple Choose Reach

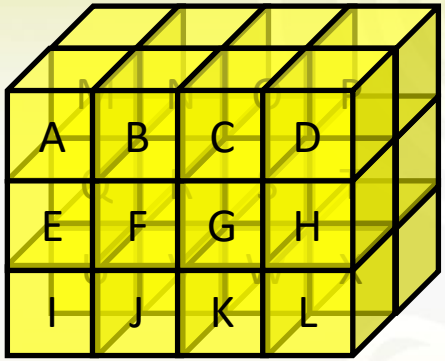
Indexed assignment `A←'YEAST' ⋄ A[3 4]←'E'`

Modified assignment `A←1 9 ⋄ A[1]+←5`

Optimised

Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⎕A
```

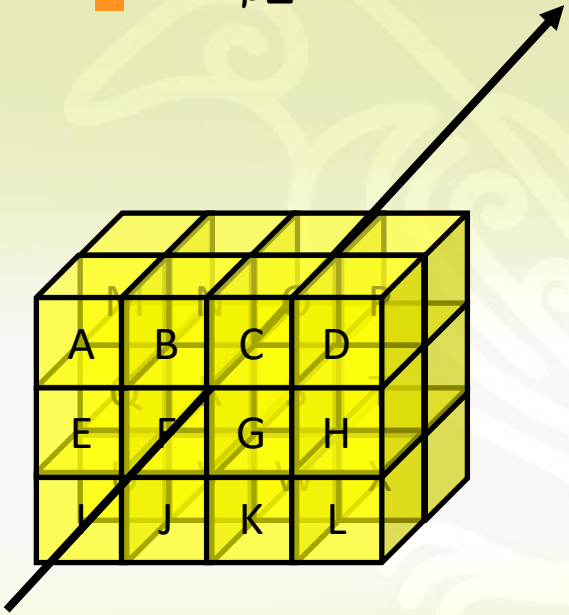


3D Array

Dimensions / Axes

Cells, subarrays and elements

```
A ← 2 3 4⍴⍳24
```

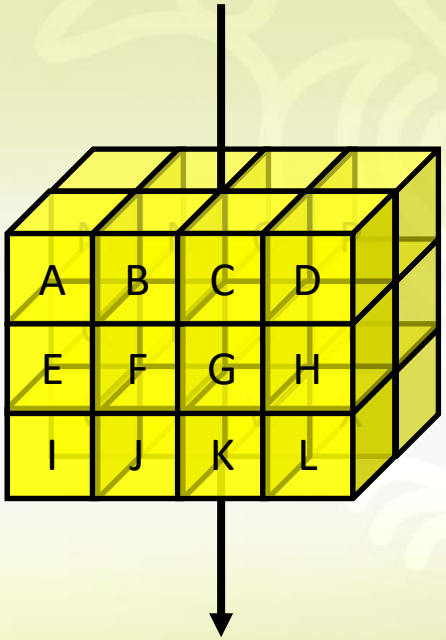


3D Array

Dimensions / Axes

Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```

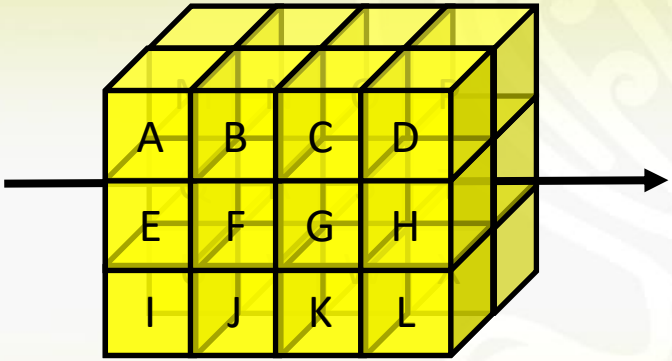


3D Array

Dimensions / Axes

Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⎕A
```



3D Array

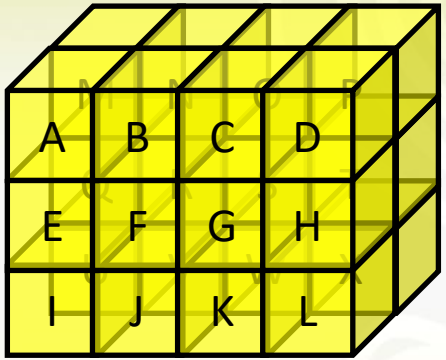
Dimensions / Axes

Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⍬
```

3D Array

Major cell: Matrix

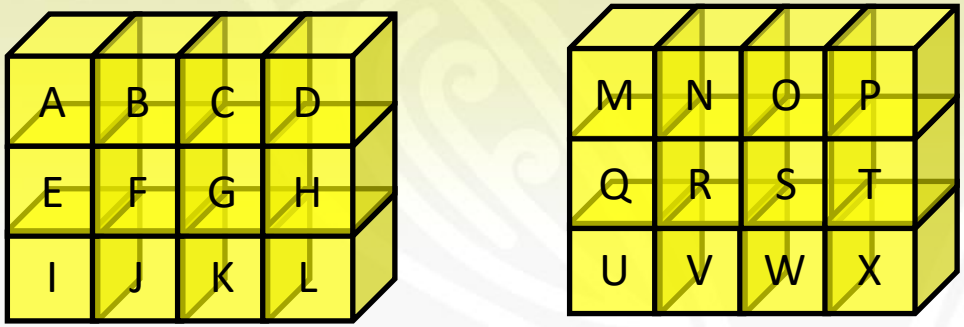


Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⎕A
```

3D Array

Major cell: Matrix

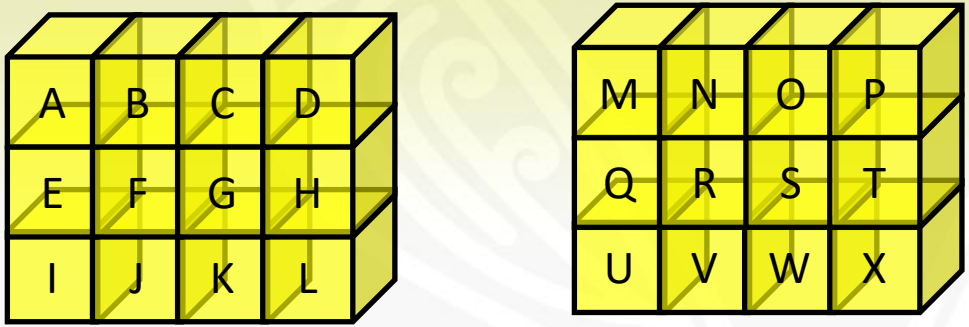


Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⍴A
```

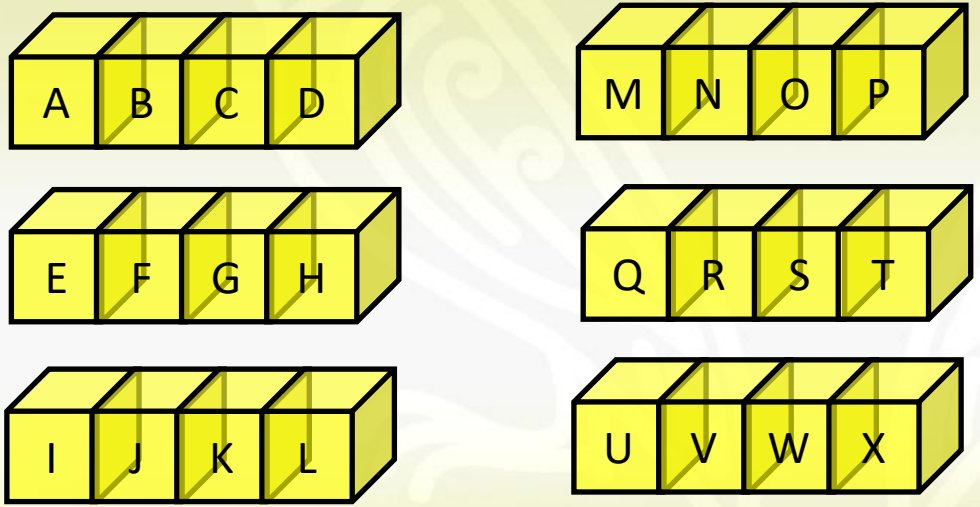
3D Array

Major cell: Matrix



Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```



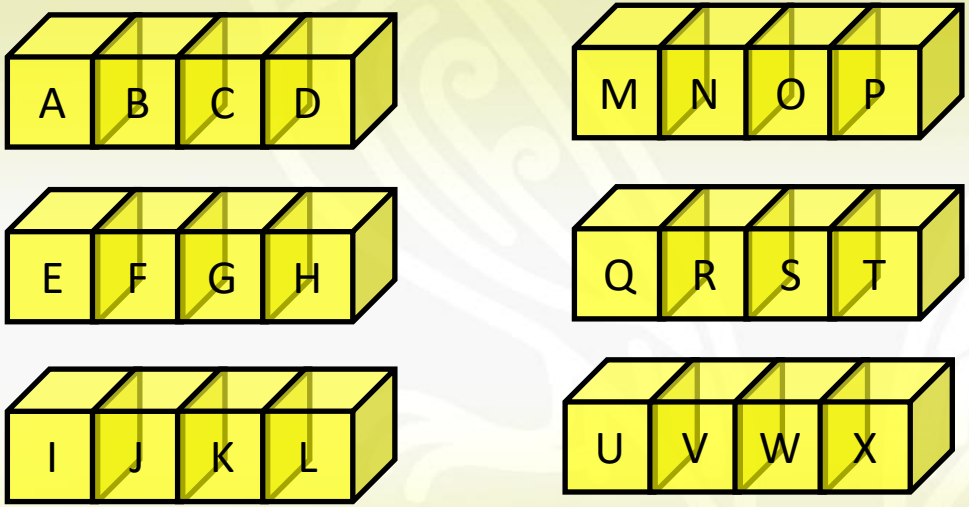
3D Array

Major cell: Matrix

1-cell: Vector

Cells, subarrays and elements

```
A ← 2 3 4⍴⍳⍞A
```



3D Array

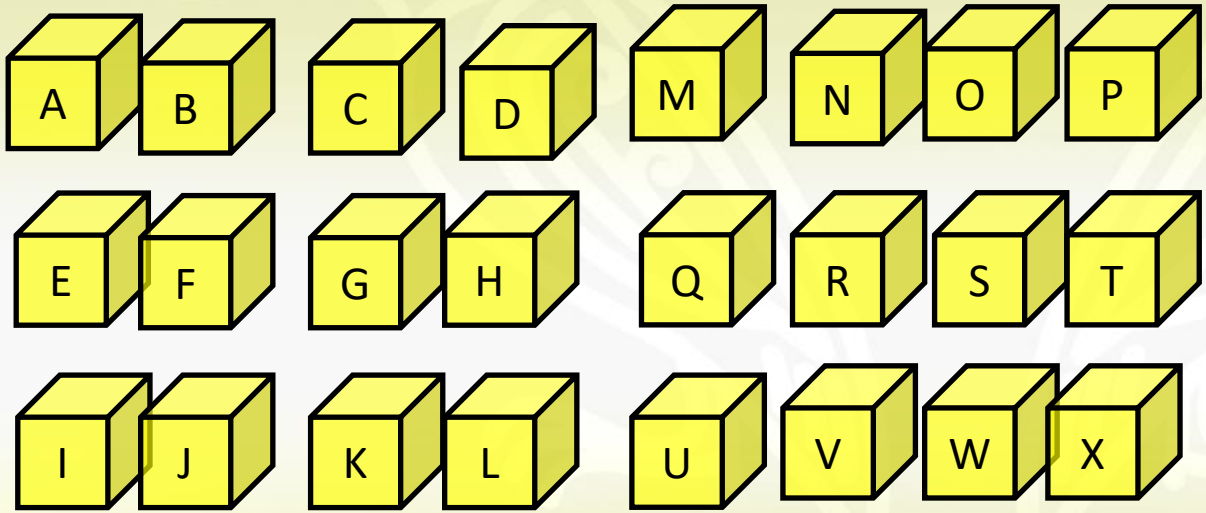
Major cell: Matrix

1-cell: Vector

0-cell: Scalar

Cells, subarrays and elements

```
A ← 2 3 4⍴⎕A
```



3D Array

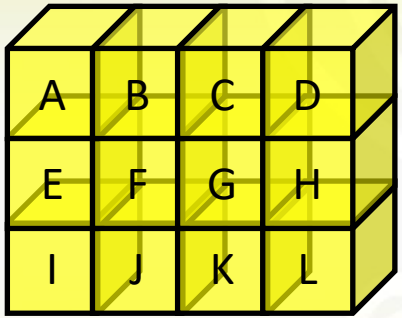
Major cell: Matrix

1-cell: Vector

0-cell: Scalar

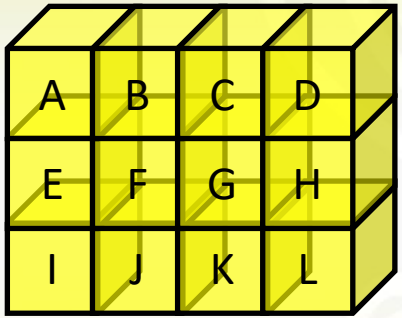
Simple indexing

```
A ← 2 3 4⍴⎕A
      A[1;;]
ABCD
EFGH
IJKL
```



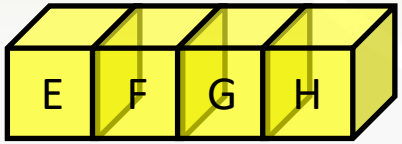
Simple indexing

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



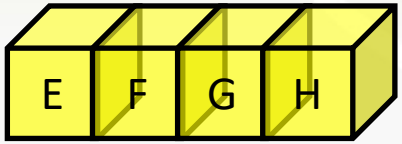
Simple indexing

```
A ← 2 3 4⍴⎕A
      A[1;2;]
EFGH
```



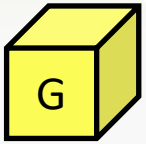
Simple indexing

```
A ← 2 3 4⍴⎕A
   1 2⌷A
EFGH
```



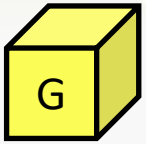
Simple indexing

```
A ← 2 3 4⍴⎕A
      A[1;2;3]
G
```



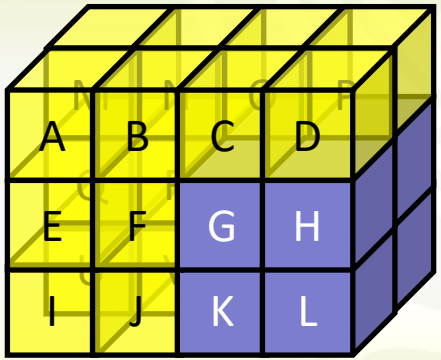
Simple indexing

```
A ← 2 3 4⍴⎕A
1 2 3⌷A
G
```



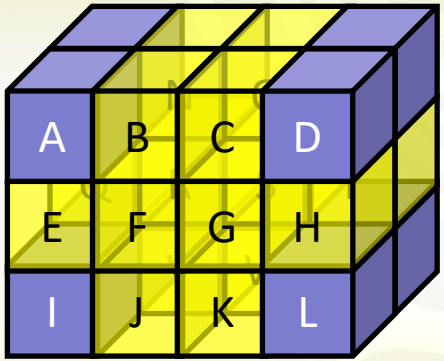
Simple indexing

```
A ← 2 3 4⍴⎕A
    A[1 2;2 3;3 4]    (1 2)(2 3)(3 4)⌷A
GH
KL

ST
WX
```



Simple indexing

```
A ← 2 3 4⍴⎕A
    A[1 2;1 3;1 4]    (1 2)(1 3)(1 4)⍉A
AD
IL

MP
UX
```



Simple indexing

```
      A ← 2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      A[1 2;1 3;1 4]
AD
IL

MP
UX
```

```
      (1 2)(1 3)(1 4)⍉A
```

Choose indexing

```
      A ← 2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      A[(1 1 1)(2 1 4)(1 3 4)]
APL
```

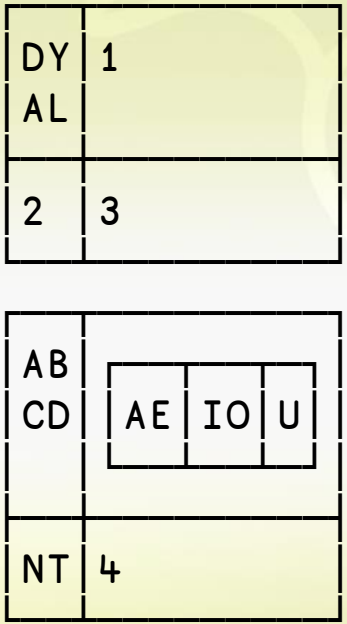
Choose indexing

```
      A ← 2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
A[(1 1 1)(2 1 4)(1 3 4)]
APL
```

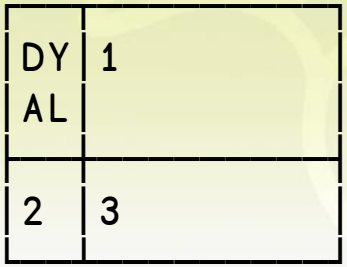
Reach indexing

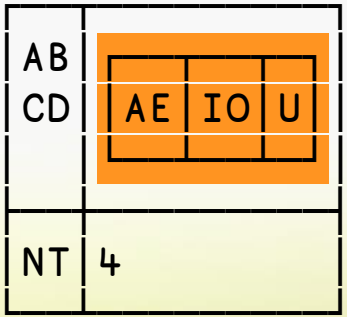
```
nest←2 2 2⍴(2 2⍴'DYAL') 1 2 3 (2 2⍴'ABCD') ('AE' 'IO' 'U') 'NT' 4
```



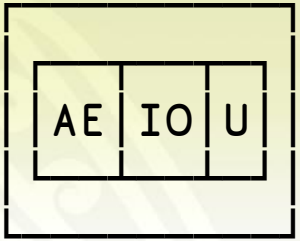
Reach indexing

nest





nest[⊂2 1 2]



⊂2 1 2

Reach indexing

nest

DY AL	1
2	3

AB CD	AE IO U
NT	4

```
nest[⊂(2 1 2)2]
```

IO

```
⊂(2 1 2)2
```

2 1 2 2

Reach indexing

nest

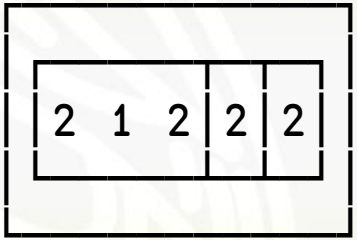
DY AL	1
2	3

AB CD	AE IO U
NT	4

```
nest[⊂(2 1 2)2(2)]
```

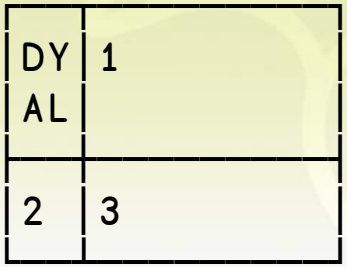
0

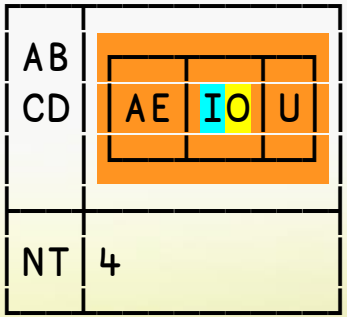
```
⊂(2 1 2)2(2)
```



Reach indexing

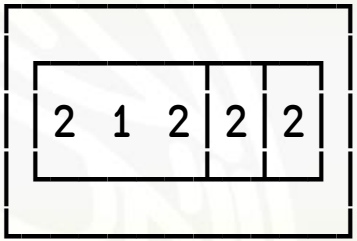
nest





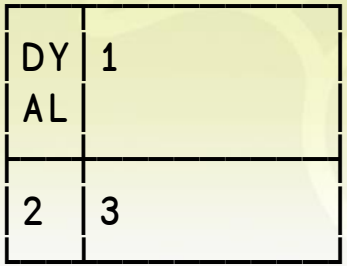

```
      DOMAIN ERROR
(⊂(2 1 2)2(2))⌷nest
               ^
```

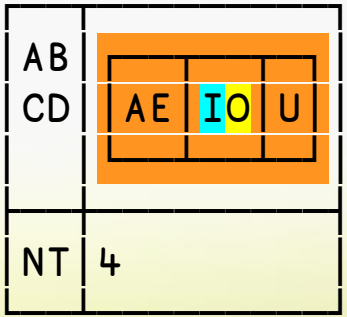
⊂(2 1 2)2(2)



Pick ~~Reach indexing~~

nest

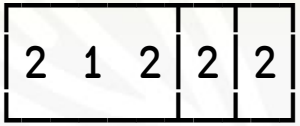





```
((2 1 2))2(2)⊃nest
```

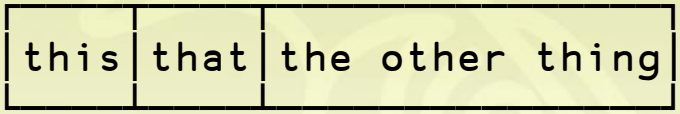
0

```
(2 1 2)2(2)
```



Pick ~~Reach indexing~~

```
      □←'this' 'that' 'the other thing'
```




```
      3⊃'this' 'that' 'the other thing'
the other thing
```

Simple

Choose

Reach

Simple **Subarrays**

Choose

Reach

Simple **Subarrays**

Choose **0-cells** AKA **Scalars**

Reach

Simple **Subarrays**

Choose **0-cells** AKA **Scalars**

Reach **Nested arrays**

Squad □

```
      A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Squad □

Indexing as a function ∴ you can use operators

```
      A←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Squad []

Indexing as a function ∴ you can use operators

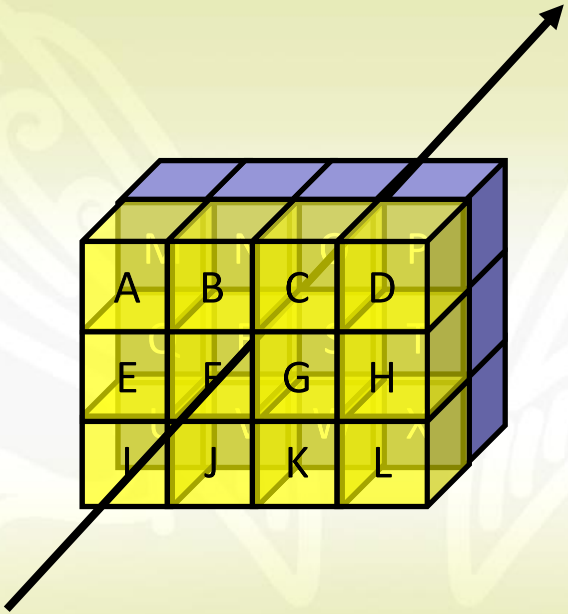
```
      A←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      2⌷⍤2⊢A
EFGH
QRST
```

```
      2⌷[2]A
```

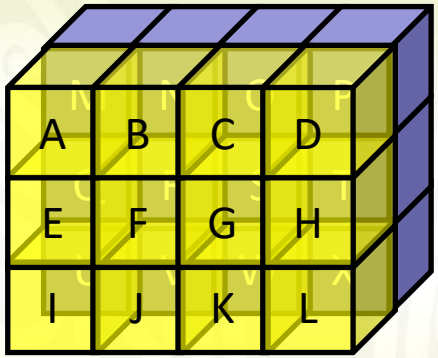
Squad with axis ⌷[ ]

```
A ← 2 3 4⍴⎕A

      2⌷[1]A
MNOP
QRST
UVWX
```



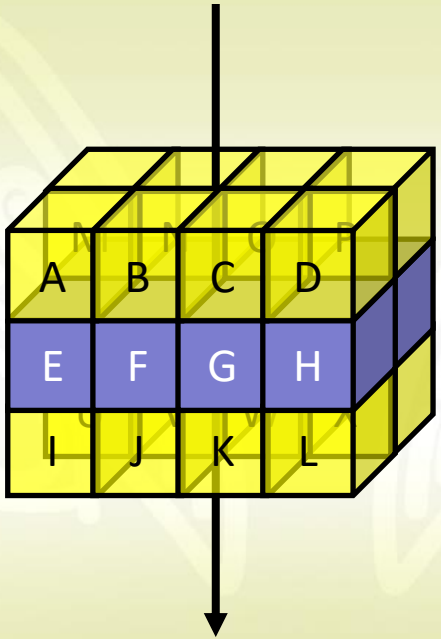
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⎕A

    2⌷⍤3⊢A
MNOP
QRST
UVWX
```



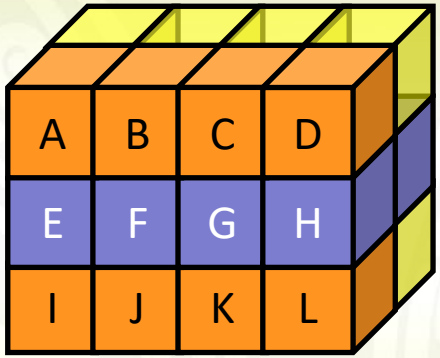
Squad with axis [ ][ ]

```
A ← 2 3 4⍴⎕A

    2[⍳][2]A
EFGH
QRST
```



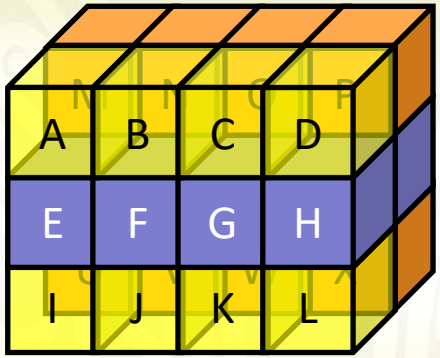
Squad with rank ⌷⍤

```
      A ← 2 3 4⍴⍳⍤⌷A

      2⌷⍤2⊢A
EFGH
QRST
```



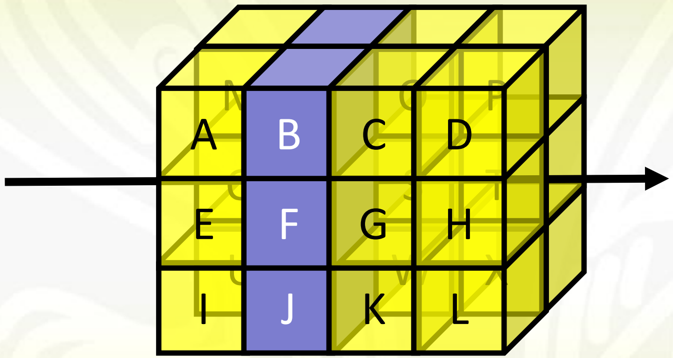
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍤A

    2⌷⍤2⊢A
EFGH
QRST
```



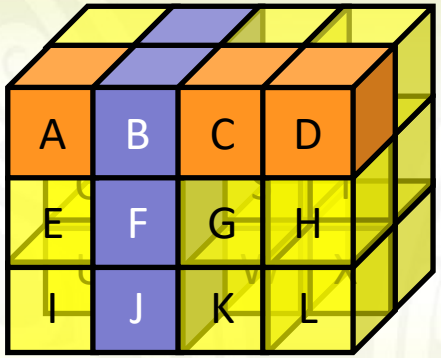
Squad with axis ⌷[ ]

```
A ← 2 3 4⍴⎕A

      2⌷[3]A
BFJ
NRV
```



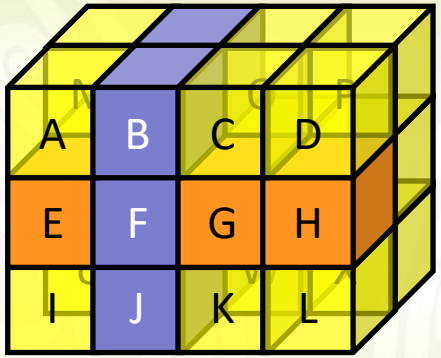
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



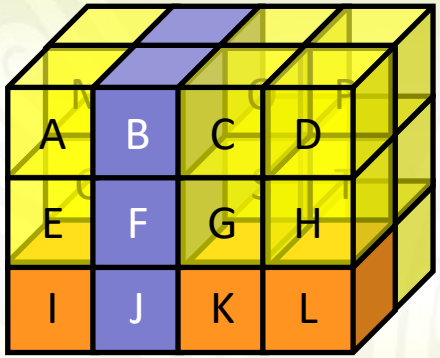
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



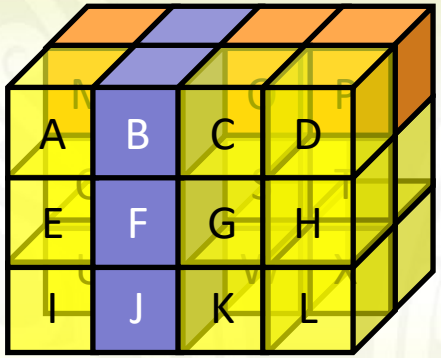
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



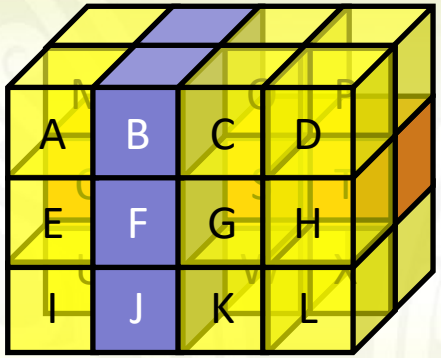
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



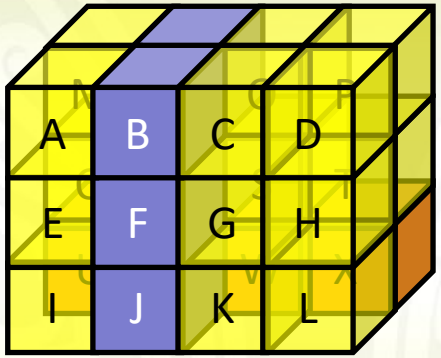
Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



Squad with rank ⌷⍤

```
A ← 2 3 4⍴⍳⍴A

    2⌷⍤1⊢A
BFJ
NRV
```



Squad with rank ⌷⍤

```
      ⌷←n←⍳3 5⍴⍳15
1  6 11
2  7 12
3  8 13
4  9 14
5 10 15
      1 3 4⌷⍤0 2⊢n
1  6 11
3  8 13
4  9 14
```

Squad with rank ⌷⍤

```
      ⎕←n←⍉3 5⍴⍳15          1 3 4⌷⍤0 99⊢n
1  6 11                1  6 11
2  7 12                3  8 13
3  8 13                4  9 14
4  9 14
5 10 15
```

"Sane" indexing

```
I ← ⎕⍠0 99
```

```
      ⎕←n←⍉3 5⍴⍳15
 1  6 11
 2  7 12
 3  8 13
 4  9 14
 5 10 15
```

"Sane" indexing

```
I ← ⎕⍠0 99
```

```
      ⎕←n←⍉3 5⍴⍳15
 1  6 11
 2  7 12
 3  8 13
 4  9 14
 5 10 15
```

```
      (⊂1 3 4)⌷n   ⍝ "Insane"
 1  6 11
 3  8 13
 4  9 14
```

"Sane" indexing

```
I ← ⎕⍠0 99
```

```
      ⎕←n←⍉3 5⍴⍳15
 1  6 11
 2  7 12
 3  8 13
 4  9 14
 5 10 15
```

```
      (⊂1 3 4)⌷n   ⍝ "Insane"
 1  6 11
 3  8 13
 4  9 14

      1 3 4 I n
 1  6 11
 3  8 13
 4  9 14
```

"Sane" indexing

```
I ← ⎕⍠0 99
```

```
      ⎕←n←⍉3 5⍴⍳15
 1  6 11
 2  7 12
 3  8 13
 4  9 14
 5 10 15
```

```
      1 3 4 I n
 1  6 11
 3  8 13
 4  9 14
      1 0 1 1 0 ≠ n
 1  6 11
 3  8 13
 4  9 14
```

"Sane" indexing

```
    'APPLE'[1 3 4]        ⍝ Intuitive, beginner friendly
APL
    1 0 1 1 0/'APPLE'     ⍝ Booleans
APL
```

"Sane" indexing

```
      1 3 4 I 'APPLE'      ⍝ Sane
APL
```

"Sane" indexing

```
      1 3 4 I 'APPLE'       ⍝ Sane
APL
      1 0 1 1 0≠'APPLE'     ⍝ Rank polymorphic
APL
```

The Indexer AKA Select

```
I ← ((⊃⊣)[]⊢)⍤0 99
```

The Indexer AKA Select

```
I ← ((⊃⊣)⌷⊢)⍤0 99
```

```
   1 I A     1⌷A     A[1;;]     1⌷⍤0 99⊢A
ABCD
EFGH
IJKL
```

The Indexer AKA Select

```
I ← ((⊃⊣)[]⊢)⍤0 99
```

```
   □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
EFGH
UVWX
```

```
(1 2)(2 3) I A
```

```
EFGH
UVWX
```

```
A[(1 2,¨2 3)∘.,⍳4]
```

```
EFGH
UVWX
```

The Indexer AKA Select

```
I ← ((⊃⊣)[]⊢)⍤0 99
```

```
   □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
H
UVWX
```

```
(1 2 4)(2 3) I A
```

The Indexer AKA Select

```
I ← ((⊃⊣)[⊢)⍤0 99
```

```
    □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
H
UVWX
```

```
(1 2 4)(2 3) I A
```

```
    r←≠⍴A
3
```

The Indexer AKA Select

```
I ← ((⊃⊣)[⊢)⍤0 99
```

```
    □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
(1 2 4)(2 3) I A
H
UVWX
```

```
    r←≠⍴A
3
    k←0
```

The Indexer AKA Select

```
I ← ((⊃⊣)[⊢)⍤0 99
```

```
    □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
H
UVWX
```

```
(1 2 4)(2 3) I A
```

```
    r←≠⍴A
3
    k←0
    r-k
3
```

The Indexer AKA Select

```
I ← ((⊃⊣)[⊢)⍤0 99
```

```
    □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
(1 2 4)(2 3) I A
H
UVWX
```

```
    r←≠⍴A
3
    k←1
```

The Indexer AKA Select

```
I ← ((⊃⊣)[⊢)⍤0 99
```

```
    □←A←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
H
UVWX
```

```
(1 2 4)(2 3) I A
```

```
    r←≠⍴A
3
    k←1
    r-k
2
```

The Indexer AKA Select

```
I ← ((⊃⊣)[]⊢)⍤0 99
      (1 2 4)(2 3) I A
H
UVWX
   NB. The "from" verb in J
   (0 1 3;1 2) { A
H
UVWX
```

The Indexer AKA Select

```
I ← ((⊃⊣)⌷⊢)⍤0 99
      1(1 3)⌷A
ABCD
IJKL
   NB. The "from" verb in J
   (<0;0 2) { A
ABCD
IJKL
```

Squad with rank ⌷⍠

```
      ⍞←A←2 3 4⍴⍞
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      (↑(1 1 1)(2 1 4)(1 3 4))⌷⍠1 99⊢A
APL
```

Squad with rank ⌷⍤

```
      ⌷←A←2 3 4⍴⌷A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      (↑(1 1 1)(2 1 4)(1 3 4))⌷⍤1 99⊢A
APL
```

```
      Choose ← ⌷⍤1 99
```

Squad with rank ⌷⍤

```
      ⎕←A←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      3⌷⍤99 2⊢A
IJKL
UVWX
      Cells ← ⌷⍤99 k
```

Squad with rank ⍠⍤

```
      ⍠←A←2 3 4⍴⍠A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      3 3⍠⍤99 2⊢A
KW
```

```
      Cells ← ⍠⍤99 k
```

Squad with rank ⌷⍤

```
      ⌷←A←2 3 4⍴⌷A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      (⊂2 3)⌷⍤99 2⊢A
EFGH
IJKL

QRST
UVWX
```

```
      Cells ← ⌷⍤99 k
```

Selecting from arrays

Selecting from arrays

[ ]

Selecting from arrays

```
[ ]
⍬
```

Selecting from arrays

```
[ ]
⍞
⊃
```

Selecting from arrays

```
[ ]
⍬
⊃
⍬[]
```

Selecting from arrays

```
[ ]
⍬
⊃
⍬[]
⍬⍤⊃
```

Selecting from arrays

```
[ ]     I ← ⍳⍥0 99 ⍴ Sane
⍸
⊃
⍸[]
⍸⍥
```

Selecting from arrays

```
[ ]     I ← ⍬⍤0 99 ⍴ Sane
⍬       I ← ((⊃⊣)⍬⊢)⍤0 99

⊃

⍬[]

⍬⍤
```

Selecting from arrays

```
[ ]     I ← ⊃⍤0 99 ⍴ Sane
⊃       I ← ((⊃⍪)⊃⊢)⍤0 99
⊃       Choose ← ⊃⍤1 99

⊃[]

⊃⍤
```

Selecting from arrays

```
[ ]
⌷
⊃
⌷[]
⌷⍤
```

```
I ← ⌷⍤0 99 ⍴ Sane
I ← ((⊃⊣)⌷⊢)⍤0 99
Choose ← ⌷⍤1 99
Cells ← ⌷⍤k
```

Performance

Performance

Pre v18.0

```
⍞⍠0 k
```

Performance

Pre v18.0

```
⎕⍠0 k
```

New in v18.0

```
⎕⍠1 99
⎕⍠99 k
```

Performance

Pre v18.0

```
      ⍞⍠0 k
```

New in v18.0

```
      ⍞⍠1 99
      ⍞⍠99 k
```

```
1=≢⍴⍺:⍞⍠k
```

Indexed assignment

```
      A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Indexed assignment

```
      A[1 2;1 3;1 4]←'ö'
ÖBCÖ
EFGH
ÖJKÖ

ÖNOÖ
QRST
ÖVWÖ
```

Search: Indexed assignment

`help.dyalog.com`

Selective assignment

```
Functions for Selective Assignment
↑       Take
↓       Drop
,       Ravel
⍪       Table
⌽⍉      Reverse, Rotate
⍴       Reshape
⊃       Disclose, Pick
⍉       Transpose (Monadic and Dyadic)
/⌿      Replicate
\⍀      Expand
[]      Index
∈       Enlist ([]ML≥1)
```

Search: Selective assignment

`help.dyalog.com`

Merge arrays with @

```
      0@((,1 3 5∘.,1 3),(2 4∘.,2))⊢n
0  6  0
2  0 12
0  8  0
4  0 14
0 10  0
```

Merge arrays with @

```
      -@(2|⊢)⊢n
¯1   6 ¯11
 2 ¯7   12
¯3   8 ¯13
 4 ¯9   14
¯5  10 ¯15
```

Selecting from arrays

```
[ ]     I ← ⍬⍤0 99 ⍴ Sane
⌷       I ← ((⊃⊣)⌷⊢)⍤0 99
⊃       Choose ← ⌷⍤1 99
⌷[]     Cells ← ⌷⍤k
⌷⍤
```

Next Week: BAA Webinar

Thursday 23rd April **16:00 BST**

Introducing qWC Alpha 0.4 – MJH Software Services

https://britishaplassociation.org/webinar-schedule-2020/

Next Dyalog Webinar

Thursday 30th April **16:00 BST**

Introducing Dyalog version 18.0 – Morten Kromberg

dyalog.tv