Language Features of version 18.0 in Depth

Adám Brudzewsky

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠Y`	Unique mask
`A⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠⍤Y`	Unique mask
`A⍨⍤⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

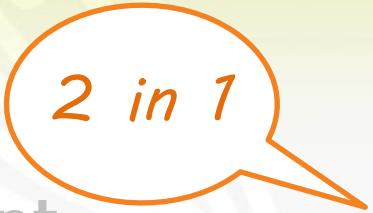
Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
```

dyalog.tv/webinar

New

⎕C	Case convert
f⍤⍥g	Over
f⍤⍤g	Atop
≠Y	Unique
A⍨	Constant
⎕DT	Date-time
1200⌶	Format date-time



Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Where on non-negative integers

Using Where

```
⍳3 0 1 4
```

History

```
PRICE←71 82 81 82 84 59
```

History

```
      PRICE←71 82 81 82 84 59
      (75≤PRICE)/PRICE
82 81 82 84
```

History

```
      PRICE←71 82 81 82 84 59
      (75≤PRICE)/PRICE
82 81 82 84
      (75≤PRICE)/⍳⍴PRICE
2 3 4 5
```

History

```
      PRICE←71 82 81 82 84 59
      (75≤PRICE)/PRICE
82 81 82 84
      (75≤PRICE)/⍳⍴PRICE
2 3 4 5
```

History

```
      PRICE←71 82 81 82 84 59
      (75≤PRICE)/PRICE
82 81 82 84
      (75≤PRICE)/⍳⍴PRICE
2 3 4 5
      ⍸75≤PRICE
2 3 4 5
```

Selection

Using Where

Use case: selection

```
fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
```

Use case: selection

```
fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
select←1 1 0 1 0
```

Use case: selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 1 0 1 0
      select/fruit
Apple  Banana  Date
```

Use case: selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 1 0 1 0
      select/fruit
 Apple  Banana  Date
      select/⍳⍴select
1 2 4
```

Use case: selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 1 0 1 0
      select/fruit
 Apple  Banana  Date
      select/⍳⍴select
1 2 4
      ⍳select
1 2 4
```

Use case: selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 1 0 1 0
      select/fruit
Apple  Banana  Date
      fruit[⍸select]
```

Use case: selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 1 0 1 0
      select/fruit
Apple  Banana  Date
      fruit[⍸select]
Apple  Banana  Date
```

Use case: multi-selection

```
fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
select←1 2 0 1 0
```

Use case: multi-selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 2 0 1 0
      select/fruit
Apple  Banana  Banana  Date
```

Use case: multi-selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 2 0 1 0
      select/fruit
 Apple  Banana  Banana  Date
      fruit[⍸select]
DOMAIN ERROR
      fruit[⍸select]
            ^
```

17·1

Use case: multi-selection

```
      fruit←'Apple' 'Banana' 'Cherry' 'Date' 'Elderberry'
      select←1 2 0 1 0
      select/fruit
Apple  Banana  Banana  Date
      fruit[⍸select]
Apple  Banana  Banana  Date
```

18·0

Use case: multi-dimensional selection

```
spice←'Anise' 'Basil' 'Chili' 'Dill' 'Epazote'
```

Use case: multi-dimensional selection

```
spice←'Anise' 'Basil' 'Chili' 'Dill' 'Epazote'
□←stuff←↑fruit spice
```

Use case: multi-dimensional selection

```
      spice←'Anise' 'Basil' 'Chili' 'Dill' 'Epazote'
      ⍞←stuff←↑fruit spice
Apple  Banana  Cherry  Date  Elderberry
Anise  Basil   Chili   Dill  Epazote
```

Use case: multi-dimensional selection

```
      spice←'Anise' 'Basil' 'Chili' 'Dill' 'Epazote'
      ⍞←stuff←↑fruit spice
Apple  Banana  Cherry  Date  Elderberry
Anise  Basil   Chili   Dill  Epazote
      ⍞←select←↑select (0 0 0 2 0)
```

Use case: multi-dimensional selection

```
      spice←'Anise' 'Basil' 'Chili' 'Dill' 'Epazote'
      ⍞←stuff←↑fruit spice
 Apple  Banana  Cherry  Date  Elderberry
 Anise  Basil   Chili   Dill  Epazote
      ⍞←select←↑select (0 0 0 2 0)
1 2 0 1 0
0 0 0 2 0
```

Use case: multi-dimensional selection

```
      stuff[⍳select]
Apple  Banana  Banana  Date  Dill  Dill
```

Use case: multi-dimensional selection

```
      stuff[⍳select]
 Apple  Banana  Banana  Date  Dill  Dill
```

```
      select/stuff
RANK ERROR
      select/stuff
             ^
```

Representing a set

Using Where

Use case: Representing a set

```
all ← 'a' 'b' 'c' 'd' 'e' 'f'
```

Use case: Representing a set

```
all  ← 'a' 'b' 'c' 'd' 'e' 'f'
mask ←  1   0   0   1   0   1
```

Use case: Representing a set

```
all     ← 'a' 'b' 'c' 'd' 'e' 'f'
mask    ←  1   0   0   1   0   1
indices ←  1               4   6
```

Use case: Representing a set

```
      all     ← 'a' 'b' 'c' 'd' 'e' 'f'
      mask    ←  1   0   0   1   0   1
      indices ←  1           4       6
      indices ≡ ⍳ mask
1
```

Use case: Representing a multi-set

```
      all     ← 'a' 'b' 'c' 'd' 'e' 'f'
      count   ←  1   0   0   3   0   2
      indices ←  1           4 4 4     6 6
      indices ≡ ⍳ count
1
```

Use case: Representing a multi-set

```
      all     ← 'a' 'b' 'c' 'd' 'e' 'f'
      count   ←  1   0   0   3   0   2
      indices ←  1           4 4 4       6 6
      indices ≡ ⍳ count
1
      count   ≡ ⍰   indices
1
```

Use case: Representing a multi-set

```
      all     ← 'a' 'b' 'c' 'd' 'e' 'f'
      count   ←  1   0   0   3   0   2
      indices ←  1           4 4 4     6 6
      indices ≡ ⍳ count
1

      count   ≡⍳⌸¯1⊢indices
1
```

Inverse Where

Whence? Whither?

```
⍸*¯1
```

Analysis of indices

Using Inverse Where

```
⍸ ⍤⍣¯1
```

Use case: Analysis of indices

```
values ← 3 14 15 35 65 89 92
```

Use case: Analysis of indices

```
values  ← 3 14 15 35 65 89 92
cutoffs ← 0 20 40 60 80 100
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⍞← bins ← cutoffs⍸values
1 1 1 2 4 5 5
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⎕← bins ← cutoffs⍸values
1 1 1 2 4 5 5
      ≠⌸⊢⍳ bins
3 1 1 2
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⎕← bins ← cutoffs⍸values
1 1 1 2 4 5 5
      ≠⌸⊢⌸ bins
3 1 1 2
      ∪ bins
1 2 4 5
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20⌶40 60 80 100⌶
      ⍞← bins ← cutoffs⍸values
1 1 1 2 4 5 5
      ≠⌸⊢⌸ bins
3 1 1 2
      ∪ bins
1 2⌶4 5⌶
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⎕← bins ← cutoffs⍸values
1 1 1 2 4 5 5
      ⍸¨⍤¯1 ⊢ bins
3 1 0 1 2
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⍞← bins ← cutoffs⍳values
1 1 1 2 4 5 5
      ⍳⍨⍣¯1 ⊢ bins
3 1 0 1 2
```

Use case: Analysis of indices

```
      values  ← 3 14 15 35 65 89 92
      cutoffs ← 0 20 40 60 80 100
      ⎕← bins ← cutoffs⍸values
1 1 1 2 4 5 5
      (≢cutoffs)↑ ⍸⍤¯1 ⊢ bins
3 1 0 1 2 0
```

Reconstruction of masks

Using Inverse Where

```
⍸⍣¯1
```

Use case: Reconstruction of masks

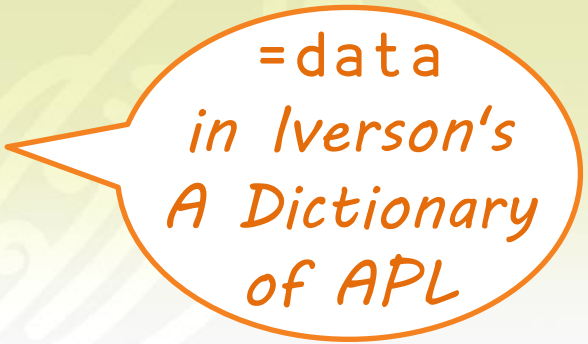
```
data ← 'Mississippi'
```

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      (∪∘.≡⊢) data
1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0 1
0 0 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0
```

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      (∪∘.≡⊢) data
1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0 1
0 0 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0
```



Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ('≡',⍪data) , data ⌸ (⍪∘.≡⊢)data
≡ M i s s i s s i p p i
M 1 0 0 0 0 0 0 0 0 0 0
i 0 1 0 0 1 0 0 1 0 0 1
s 0 0 1 1 0 1 1 0 0 0 0
p 0 0 0 0 0 0 0 0 1 1 0
```

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      (('≡',∪) , ⊢ ⍪ ∪∘.≡⊢)data
≡ M i s s i s s i p p i
M 1 0 0 0 0 0 0 0 0 0 0
i 0 1 0 0 1 0 0 1 0 0 1
s 0 0 1 1 0 1 1 0 0 0 0
p 0 0 0 0 0 0 0 0 1 1 0
```

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ⊢⌸ data
1  0 0  0
2  5 8 11
3  4 6  7
9 10 0  0
```

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ⊢⌸ data
1  0 0  0
2  5 8 11
3  4 6  7
9 10 0  0
```

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ⊢⌸ data
1  0 0  0
2  5 8 11
3  4 6  7
9 10 0  0
      ⊂⍤⊢⌸ data
```

1	2 5 8 11	3 4 6 7	9 10

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ⊢⌸ data
1  0 0  0
2  5 8 11
3  4 6  7
9 10 0  0
      ⍳⍤≠⍥⊂1⍥⊢⌸ data
1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0 1
0 0 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0
```

=data
in Iverson's
A Dictionary
of APL

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ⊢⌸ data
1  0 0  0
2  5 8 11
3  4 6  7
9 10 0  0
      ⍳⍤≠⍨1⍥⊢⌸ data
1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 1 0 0 1
0 0 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0
```

=data in Iverson's A Dictionary of APL

Use case: Reconstruction of masks

```
data ← 'Mississippi'
]runtime -compare  (∪⍤.≡⊢)data  ⍸⍤¯1⍤⊢⌸data
```

Use case: Reconstruction of masks

```
      data ← 'Mississippi'
      ]runtime -compare  (∪∘.≡⊢)data  ⍸⍣¯1⍤⊢⌸data

(∪∘.≡⊢)data → 3.8E¯6 |   0% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
⍸⍣¯1⍤⊢⌸data → 3.2E¯6 | -18% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
```

Use case: Reconstruction of masks

```
data ← ?1000⍴100
]runtime -compare  (∪⍤.≡⊢)data  ⍳⍤¯1⍤⊢⌸data
```

Use case: Reconstruction of masks

```
      data ← ?1000⍴100
      ]runtime -compare  (∪∘.≡⊢)data  ⍸⍣¯1⍥⊢⌸data

(∪∘.≡⊢)data → 4.2E¯3 |   0% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
⍸⍣¯1⍥⊢⌸data → 5.5E¯5 | -99% ⎕
```

New

```
⎕C      Case convert
f⍤⍥g    Over
f⍤⍤g    Atop
≠⌸Y     Unique mask
A⍨⍨     Constant
⎕DT     Date-time
1200⌶   Format date-time
```

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NGET⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Questions?

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠Y`	Unique mask
`A⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Advanced Use of The Rank Operator

Richard Park