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⍠'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⍠'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⍠'Regex'
⍁NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y

dyalog.tv/webinar

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⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y

Constant

makes `{0}` and `{0}¨` obsolete

A≍

Why?

Lightweight notation

Proper visual order in trains

Avoid ugly work-arounds

Lightweight notation

```
      'Hi'(1 2 3⍨)'Earth'
1 2 3
      (1 2 3⍨)'Hello'
1 2 3
      # ⎕SE⍨'World'
# ⎕SE
```

Lightweight notation

Train	`{A}×h`
At	`{A}@h`

Lightweight notation

Train	`{A}×h`	`A×h`
At	`{A}@h`	`A@h`

Lightweight notation

Train	`{A}×h`	`A×h`
At	`{A}@h`	`A@h`
Constant	`{A}`	

Lightweight notation

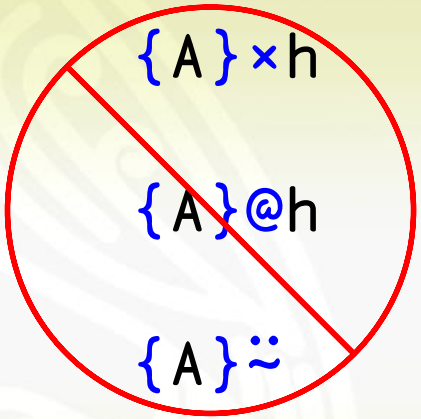
Train	`{A}×h`	`A×h`
At	`{A}@h`	`A@h`
Constant	`{A}¨⍨`	

Lightweight notation

Train	`{A}×h`	`A×h`
At	`{A}@h`	`A@h`
Constant	`{A}⍨`	`A⍨`

Lightweight notation

Train	{A}×h	A×h
At	{A}@h	A@h
Constant	{A}⍨	A⍨



Lightweight notation

```
      42⍨⍨⍨'Life' 'Universe' 'Everything'
42 42 42
```

Lightweight notation

```
      42⍤⍤'Life' 'Universe' 'Everything'
42 42 42
      (?9)⍤⍤1 2∘.×1 2 3 4
7 7 7 7
7 7 7 7
```

Lightweight notation

```
      42⍤⍤⊢'Life' 'Universe' 'Everything'
42 42 42
      (?9)⍤⍤⊢1 2∘.×1 2 3 4
7 7 7 7
7 7 7 7
      ('R'⍤⍤⍥1)2 3 4⍴⍳24
RRR
RRR
```

Proper visual order in trains

normal code `(⍺+⍵)*3`

Proper visual order in trains

normal code `(⍺+⍵)*3`

invalid train `+*3`

Proper visual order in trains

normal code	`(⍺+⍵)*3`
invalid train	`+*3`
workaround	`3*⍨+`
workaround	`*∘3+`

Proper visual order in trains

normal code `(α+ω)*3`

invalid train `+*3`

valid train `+*3⍨`

Proper visual order in trains

normal code `(⍺+⍵)*÷3`

Proper visual order in trains

normal code `(⍺+⍵)*÷3`

invalid train `+*∘÷3`

Proper visual order in trains

normal code	`(⍺+⍵)*÷3`
invalid train	`+*∘÷3`
workaround	`(÷3)*⍨+`
workaround	`3*∘÷⍨+`

Proper visual order in trains

normal code	`(⍺+⍵)*÷3`
invalid train	`+*∘÷3`
valid train	`+*∘÷3⍨`
valid train	`+*1÷3⍨`

Avoid ugly work-arounds

```
mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'


      (mask/data)←'□' ⋄ data
□bcd□fg
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
      '□'@{mask}data
□bcd□fg
      (mask/data)←'□' ⋄ data
□bcd□fg
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
      '□'@{mask}data
□bcd□fg
      mask{'□'@{α}ω}data
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
      '□'@{mask}data
□bcd□fg
      mask{'□'@{α}ω}data
VALUE ERROR
      mask{'□'@{α}ω}data
                 ^
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
      '□'@{mask}data
□bcd□fg
      mask{'□'@{α}ω}data
VALUE ERROR
      mask{'□'@{α}ω}data
                ^
```

Avoid ugly work-arounds

```
      mask←1 0 0 0 1 0 0 ⋄ data←'AbcdEfg'
      '□'@{mask}data
□bcd□fg
      mask{'□'@(α⍨)ω}data
□bcd□fg
```

Enough, give it to me already!

Enough, give it to me already!

```
Constant ← {⍺⍺}
```

Date & time

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⍠'Regex'
⎕NPUT⍠'NEOL'
⍸Y
X⊂Y
↑[k]Y
```

Date-time

Timestamp

Time number

Military time zone

Timestamp

year month day... ms `2020 6 11 16 0 0 0`

year week weekday... µs `2020 24 4 16 0 0 0`

⎕TS

Timestamp

year month day... ms 2020 6 11 16 0 0 0 ←BST— ⍬TS

year week weekday... µs 2020 24 4 16 0 0 0

Time number

days since 1899-12-31 00:00 `43992.70833` ←BST

seconds since 1970-01-01 00:00 `1591894800`

Military time zone

```
'Y'  UTC-12
 ⋮
'P'  UTC-3
'O'  UTC-2
'N'  UTC-1
'Z'  UTC
'A'  UTC+1
'B'  UTC+2
'C'  UTC+3
 ⋮
'M'  UTC+12
```

```
'J'  local
```

Military time zone

```
'Y'  UTC-12
 ⋮
'P'  UTC-3
'O'  UTC-2
'N'  UTC-1
'Z'  UTC
'A'  UTC+1   ←BST—   'J'  local
'B'  UTC+2
'C'  UTC+3
 ⋮
'M'  UTC+12
```


Military time zone
Y X W V U T S R Q P O N Z A B C D E F G H I K L M
ARCTIC OCEAN
Greenland
(DENMARK)
CANADA
U.S.
UNITED STATES
MEXICO
NORTH PACIFIC OCEAN
NORTH ATLANTIC OCEAN
SOUTH PACIFIC OCEAN
SOUTH ATLANTIC OCEAN
INDIAN OCEAN
RUSSIA
CHINA
MONGOLIA
KAZAKHSTAN
IRAN
INDIA
INDONESIA
AUSTRALIA
BRAZIL
ARGENTINA
CHILE
PERU
BOLIVIA
PARAGUAY
COLOMBIA
VENEZUELA
ECUADOR
NIGERIA
ALGERIA
LIBYA
EGYPT
SUDAN
CHAD
NIGER
MALI
MAURITANIA
ETHIOPIA
KENYA
TANZANIA
ANGOLA
NAMIBIA
SOUTH AFRICA
ZAMBIA
MOZAMBIQUE
MADAGASCAR
SAUDI ARABIA
YEMEN
IRAQ
TURKEY
SYRIA
FINLAND
SWEDEN
NORWAY
UNITED KINGDOM
IRELAND
FRANCE
SPAIN
GERMANY
POLAND
UKRAINE
BELARUS
JAPAN
PHILIPPINES
MALAYSIA
NEW ZEALAND
PAPUA NEW GUINEA
French Polynesia
(FRANCE)
International Date Line
Coordinated Universal Time (UTC)
formerly Greenwich Mean Time (GMT)


Military time zone
Y X W V U T S R Q P O N Z A B C D E F G H I K L M
J
Greenland
(DENMARK)
CANADA
U.S.
UNITED STATES
MEXICO
RUSSIA
CHINA
MONGOLIA
KAZAKHSTAN
INDIA
INDONESIA
AUSTRALIA
BRAZIL
ARGENTINA
CHILE
PERU
BOLIVIA
COLOMBIA
VENEZUELA
SOUTH AFRICA
NIGERIA
SUDAN
LIBYA
ALGERIA
FRANCE
SPAIN
GERMANY
NORWAY
SWEDEN
FINLAND
UNITED KINGDOM
IRELAND
ICELAND
NEW ZEALAND
PHILIPPINES
JAPAN
ARCTIC OCEAN
NORTH PACIFIC OCEAN
NORTH ATLANTIC OCEAN
SOUTH PACIFIC OCEAN
SOUTH ATLANTIC OCEAN
INDIAN OCEAN
International Date Line
Coordinated Universal Time (UTC)
formerly Greenwich Mean Time (GMT)

Time numbers

1	Dyalog day number
2	Dyalog component file
10	J nanoseconds
11	Shakti K milliseconds
12	JavaScript/D/Q ms
13	R chron format
20	Unix time
30	MS-DOS date/time
31	MS-Win32 FILETIME

et cetera ad abundantiam

Timestamps

¯1	⎕TS-style: year month... ms
¯2	Like ⎕TS but µs replacing ms
¯3	Like ⎕TS but ns replacing ms
¯10	ISO year day hour min sec µs
¯11	ISO year week weekday... µs

etc.

⎕DT syntax

Conversion

```
          outCode ⎕DT dateTimes
inCode outCode ⎕DT dateTimes
```

Validation

```
          0 ⎕DT dateTimes
inCode 0 ⎕DT dateTimes
```

⎕DT syntax: one-element left argument

Conversion: ⎕TS-style timestamp to Unix time

```
      20 ⎕DT ⊂ 2020 06 11  16 00 00 000
1591891200
```

Validation: Leap year check

```
      0 ⎕DT ⊂ 1900 02 29
0
```

⎕DT syntax: one-element left argument

Conversion: Current ⎕TS-style UTC time

```
      ¯1 ⎕DT 'Z'
 2020 6 11 16 0 0 0
```

What time zone am I in?

```
      3600÷⍨-/20 ⎕DT 'JZ'
1
```

⎕DT syntax: one-element left argument

Conversion: Current ⎕TS-style UTC time

```
      ¯1 ⎕DT 'Z'
2020 6 11 16 0 0 0
```

What time zone am I in?

```
      3600÷⍨-/20 ⎕DT 'JZ'
1
```

←BST=UTC+1

⎕DT syntax: two-element left argument

Conversion: Unix time to ⎕TS-style timestamp

```
      20 ¯1 ⎕DT 1591891200
2020 6 11 16 0 0 0
```

Validation: Leap year check

```
      60 0 ⎕DT 19000229
0
```

1200I syntax

Format one or more date-times

```
format (1200I) dyalogDateNumbers
```

1200I syntax

Format one or more date-times

```
format (1200I) dyalogDateNumbers
format (1200I) code 1 ⎕DT dateTimes
```

1200I syntax

Format one or more date-times

```
format (1200I) dyalogDateNumbers
format (1200I) code 1 ⎕DT dateTimes
```

1200I syntax

Format one or more date-times

```
format (1200I) dyalogDateNumbers
format (1200I) code 1 ⎕DT dateTimes
```

Wikidata item

Languages

한국어

Edit links

Print/export

Download as PDF

Printable version

11, 2006" in MDY, and "2010 November 6" in YMD.

The ISO 8601 format YYYY-MM-DD (2020-06-04) is intended to harmonize these formats and ensure accuracy in all situations. Many countries have adopted it as their sole official date format, though even in these areas writers may adopt abbreviated formats that are no longer recommended.

Table coding [edit source]

Basic components of a calendar date for the most common calendar systems:

Y – year
M – month
D – day

Specific formats for the basic components:

yy – two-digit year, *e.g. 06*
yyyy – four-digit year, *e.g. 2006*
m – one-digit month for months below 10, *e.g. 4*
mm – two-digit month, *e.g. 04*
mmm – three-letter abbreviation for month, *e.g. Apr*
mmmm – month spelled out in full, *e.g. April*
d – one-digit day of the month for days below 10, *e.g. 2*
dd – two-digit day of the month, *e.g. 02*

1200I syntax

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

1200I syntax: patterns

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

1200I syntax: patterns

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

```
'M'       '6'
'MM'      '06'
'MMM'     'JUN'
'MMMM'    'JUNE'
```

1200I syntax: numbers

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

```
'M'      '6'
'MM'     '06'
'_M'     ' 6'
```

1200I syntax: names

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

```
'MMMM'          'JUNE'
'__en__MMMM'    'JUNE'
'__ru__MMMM'    'ИЮНЬ'
'__fr__MMMM'    'JUIN'
```

1200I syntax: names

Format one or more date-times

```
'YYYY-MM-DD' (1200I) dyalogDateNumbers
```

```
'mmmm'    'june'
'Mmmm'    'June'
'MMMM'    'JUNE'
```

1200I syntax: names

Format one or more date-times

```
      'YYYY-MM-DD' (1200I) dyalogDateNumbers
                  __en__   __fr__

 'mmmm'    'june'   'juin'
 'Mmmm'    'June'   'Juin'
 'MMMM'    'JUNE'   'JUIN'
```

1200I syntax: names

Format one or more date-times

```
      'YYYY-MM-DD' (1200I) dyalogDateNumbers
                __en__    __fr__
'_mmm'    'June'    'juin'
'mmmm'    'june'    'juin'
'Mmmm'    'June'    'Juin'
'MMMM'    'JUNE'    'JUIN'
```

1200I syntax: ordinals

Format one or more date-times

```
      'YYYY-MM-DD' (1200I) dyalogDateNumbers
              __en__   __fr__
'D'           '1'      '1'
'Doo'         '1st'    '1er'
'DD'          '11'     '11'
'DDoo'        '11th'   '11'
```

1200I syntax: ordinals

Format one or more date-times

```
      'YYYY-MM-DD' (1200I) dyalogDateNumbers
              __en__   __da__
'D'           '1'      '1'
'Doo'         '1st'    '1.'
'DD'          '11'     '11'
'DDoo'        '11th'   '11.'
```

1200I syntax: 12/24 hours

Format one or more date-times

```
'hh:mm' (1200I) dyalogDateNumbers
```

```
'h'       '16'
't'       '4'
't pp'    '4 pm'
'tPP'     '4PM'
```

1200I examples

```
      'DDoo Mmmm YYYY "at" hh:mm:ss.fff'
11th June 2020 at 16:00:00.000
      '__da__Dddd, D. mmmm "'"YY'
Torsdag, 11. juni '20
      '%ISO%'
2020-06-11T16:00:00
```

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⍠'Regex'
⎕NPUT⍠'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⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Thinking in APL: Array-Oriented Solutions

Richard Park

Part 1
June 25, 2020