

ipynb_onself

July 24, 2022

0.0.1 :::jword::: IFACEWORDSipynb

```
[ ]: NB. interface words (IFACEWORDSipynb) group
IFACEWORDSipynb=:,<'ipynbfrjod'
```

0.0.2 :::jword::: JWORDMARK

```
[ ]: NB. prefix for markdown sections - converts to easily found strings in notebooks
JWORDMARK=: ' :::jword::: '
```

0.0.3 :::jword::: LF

```
[ ]: NB. line feed character
LF=:10{a.
```

0.0.4 :::jword::: MDSECTION

```
[ ]: NB. markdown cell section marker
MDSECTION=: '###'
```

0.0.5 :::jword::: NBHEADER

```
[ ]: NBHEADER=:123 10 32 34 99 101 108 108 115 34 58 32 91 10{a.
```

0.0.6 :::jword::: NBJCELLBEGen

```
[ ]: NBJCELLBEGen=:32 32 32 93 10 32 32 125 44 10 32 32 123 10 32 32 32 34 99 101_
↪108 108 95 116 121 112 101 34 58 32 34 99 111 100 101 34 44 10 32 32 32 34_
↪101 120 101 99 117 116 105 111 110 95 99 111 117 110 116 34 58 32 110 117_
↪108 108 44 10 32 32 32 34 109 101 116 97 100 97 116 97 34 58 32 123 125 44_
↪10 32 32 32 34 111 117 116 112 117 116 115 34 58 32 91 93 44 10 32 32 32 34_
↪115 111 117 114 99 101 34 58 32 91 10{a.
```

0.0.7 :::jword::: NBJCELLBEGst

```
[ ]: NBJCELLBEGst=:32 32 123 10 32 32 32 34 99 101 108 108 95 116 121 112 101 34 58␣  
  ↪32 34 109 97 114 107 100 111 119 110 34 44 10 32 32 32 34 109 101 116 97 100␣  
  ↪97 116 97 34 58 32 123 125 44 10 32 32 32 34 115 111 117 114 99 101 34 58 32␣  
  ↪91 10{a.
```

0.0.8 :::jword::: NBJCELLEND

```
[ ]: NBJCELLEND=:32 32 32 93 10 32 32 125 44 10{a.
```

0.0.9 :::jword::: NBTRAILER

```
[ ]: NBTRAILER=:93 44 10 32 34 109 101 116 97 100 97 116 97 34 58 32 123 10 32 32 34␣  
  ↪107 101 114 110 101 108 115 112 101 99 34 58 32 123 10 32 32 32 34 100 105␣  
  ↪115 112 108 97 121 95 110 97 109 101 34 58 32 34 74 34 44 10 32 32 32 34 108␣  
  ↪97 110 103 117 97 103 101 34 58 32 34 74 34 44 10 32 32 32 34 110 97 109 101␣  
  ↪34 58 32 34 106 107 101 114 110 101 108 34 10 32 32 125 44 10 32 32 34 108␣  
  ↪97 110 103 117 97 103 101 95 105 110 102 111 34 58 32 123 10 32 32 32 34 102␣  
  ↪105 108 101 95 101 120 116 101 110 115 105 111 110 34 58 32 34 46 105 106␣  
  ↪115 34 44 10 32 32 32 34 109 105 109 101 116 121 112 101 34 58 32 34 116 101␣  
  ↪120 116 47 74 34 44 10 32 32 32 34 110 97 109 101 34 58 32 34 74 34 10 32 32␣  
  ↪125 10 32 125 44 10 32 34 110 98 102 111 114 109 97 116 34 58 32 52 44 10 32␣  
  ↪34 110 98 102 111 114 109 97 116 95 109 105 110 111 114 34 58 32 53 10 125␣  
  ↪10{a.
```

0.0.10 :::jword::: PYESCAPECHRS

```
[ ]: NB. common python string escape characters  
PYESCAPECHRS=:254 92 39 254 39 254 92 92 254 92 254 92 34 254 34{a.
```

0.0.11 :::jword::: REVPYESCAPECHRS

```
[ ]: NB. reverse python escapes - excluding single quote - order matters  
REVPYESCAPECHRS=:254 92 254 92 92 254 34 254 92 34{a.
```

0.0.12 :::jword::: ROOTWORDSiptynb

```
[ ]: NB. root words (ROOTWORDSiptynb) group  
ROOTWORDSiptynb=:<;._1 ' IFACEWORDSiptynb PYESCAPECHRS ROOTWORDSiptynb VMDiptynb␣  
  ↪iptynbfrjod'
```

0.0.13 :::jword::: VMDipynb

```
[ ]: NB. version, make count and date
VMDipynb=: '0.8.0';5;'24 Jul 2022 13:25:21'
```

0.0.14 :::jword::: beforelaststr

```
[ ]: NB. retains string (y) before last occurrence of (x)
beforelaststr=:] {.~ 1&(i:~)@([ E. ])
```

0.0.15 :::jword::: changestr

```
[ ]: changestr=:4 : 0

NB.*changestr v-- replaces substrings - see long documentation.
NB.
NB. dyad: clReps changestr cl
NB.
NB. NB. first character delimits replacements
NB. '/change/becomes/me/ehh' changestr 'blah blah ...'

pairs=. 2 {. "(1) _2 [\ <;._1 x      NB. change table
cnt=. _1 [ lim=. # pairs
while. lim > cnt=.>:cnt do.          NB. process each change pair
  't c'=. cnt { pairs                NB. /target/change
  if. +./b=. t E. y do.              NB. next if no target
    r=. I. b                         NB. target starts
    'l q'=. #&> cnt { pairs           NB. lengths
    p=. r + 0,+/\(<:# r)$ d=. q - 1  NB. change starts
    s=. * d                          NB. reduce < and > to =
    if. s = _1 do.
      b=. 1 #~ # b
      b=. ((1 * # r)$ 1 0 #~ q,l-q) (,r +/ i. 1)} b
      y=. b # y
      if. q = 0 do. continue. end.   NB. next for deletions
    elseif. s = 1 do.
      y=. y #~ >: d r} b             NB. first target char replicated
    end.
    y=. (c $~ q *# r) (,p +/i. q)} y  NB. insert replacements
  end.
end. y                                NB. altered string
)
```

0.0.16 :::jword::: dblquote

```
[ ]: NB. enclose all character lists in blcl in " quotes
dblquote=: ' "'&, @:(, &' "' )&.>
```

0.0.17 :::jword::: ipynbfrjod

```
[ ]: ipynbfrjod=:3 : 0

NB.*ipynbfrjod v-- extract J words from JOD and insert in blank
NB. jupyter notebook.
NB.
NB. monad: clIpynb =. ipynbfrjod blclNames
NB.
NB. NB. examples use docs and utils
NB. require 'general/jod'
NB. od ;:'docs utils'
NB.
NB. nbj=: ipynbfrjod ;:'sha1 sha1dir'
NB. nbj write 'C:\Users\baker\jupyter_notebooks\test0.ipynb'
NB.
NB. nbj=: ipynbfrjod }. grp 'ipynb'
NB. nbj write 'C:\Users\baker\jupyter_notebooks\ipynb_onself.ipynb'

NB. require 'general/jod' !(*)=. disp
jc=. disp&.> y

NB. markdown sections with word name
sec=. dblquote (<MDSECTION,JWORDMARK) ,&.> y
sec=. (<NBCELLBEGst) ,&.> sec ,&.> <NBCELLBEGen

NB. j code to quoted list of python strings notebook format
nbj=. <;._2@ (REVPYESCAPECHRS&changestr)@tlf&.> jc
nbj=. ;&.> ' "' , L: 0 (<'\" , LF) ,~ L: 0 nbj
nbj=. ,&' "'&.> '\" , '&beforelaststr&.> nbj
nbj=. sec ,&.> nbj ,&.> <NBCELLEND
toJ NBHEADER , (LF ,~ ' , '&beforelaststr ;nbj) , NBTRAILER
)
```

0.0.18 :::jword::: tlf

```
[ ]: NB. appends trailing line feed character if necessary
tlf=:] , ((10{a.)"_ = {:) }. (10{a.)"_
```

0.0.19 :::jword::: toJ

```
[ ]: NB. converts character strings to J delimiter LF  
toJ=:((10{a.) I.@(e.&(13{a.))@]} ]>@:(#~ -.@((13 10{a.)&E.@,))
```

0.0.20 :::jword::: write

```
[ ]: NB. writes a list of bytes to file  
write=:1!:2 ]`<@.(32&>@.(3!:0))
```