00_MyQCheatSheet

September 22, 2019

1 My Q cheat sheet

1.1 Check if infinit value

[ ]:
```
0w in (-0W; 0W) // throws error 'type
```

[ ]:
```
myVar:0w
(myVar ~ 0W) or (myVar ~ -0W) // works
```

Create empty list

[ ]:
```
0#0 ~ `long$()
0#0.0 ~ `float$()
```

1.2 File I/O

- Binary file handling of q data structures as content
  - x set y:
    - x is a binary file handle including its name
    - y is a q data structure
    - saves y as x
    - file name can be customized
  - get x:
    - x is a binary file handle
    - reads x into memory as a q data structure
  - save x (similar to set[x;y]);
    - x is a filename as a symbol
    - x (ignoring path and extension) must match the name of a global table
    - saves the global table to file and returns the filename
    - the format used depends on the file extension: **.txt, .csv, .xls, .xml**
    - file name cannot be customized
    - e.g.: *save hsym ‘global_table_name*
    - to customize file name use this idiom:
      - “file_name.csv 0: csv 0: table
      - saves table to .csv file while controlling the filename
      - path cannot be customized: file is saved in HOME directory

  - load x (similar to get x):
    - x is a filehandle
    - reads x and creates q data structure persisted in the file with the same name as the input filename
- Binary file handling of bites as content
  - 1: x
    - x is raw binary data
    - writes x to binary file
    - file name cannot be customized
  - read1 x:
    - x is a binary file handle
    - reads contents of x into memory as a list of bites
- Text file handling: .txt, .csv, .tsv
  - 0: x
    - x is a list of strings
    - writes x into text file format (txt, csv, tsv)
    - path cannot be customized
  - read0 x:
    - x is a text file handle
    - reads contents of x into memory as a list of strings
- File handling related built-on functions:
  - hcount x: Returns as a long integer equal to the size in bytes of x
  - hsym x:
    - x is symbol containing a path ending in a file name
    - hsym convers x into a file handle by inserting a colon between the backtick and path
    - e.g.: hsym `fdf/df.txt ->`:fdf/df.txt
    - hdel x: deletes file by handle name
  - hopen x:
    - x is a file or a websocket handle
    - returns an open handle function (creates function if it does not exist)
  - hclose x: closes the file or websocket handle
  - neg h l:
    - h is an open file handle function
    - l is a list of strings
    - appends items of l to h
    - e.g.: neg h (value1;value2)

1.3 Idiotic one character functions

- Ternary / conditional expression:
  - $[cond_expr;true_expr;false_expr]
- Exception handling (protected evaluation)
  - @[monadic_function;single_parameter;'ErrorMessage]
  - .[multivalent_function;list_of_parameters;'ErrorMessage]
- Functional select:
  - ?[table;constraint;byGroup;aggregates;limit;orderBy] // select and exec when rank is 4 or above

  - ?[x;y;z] // vector conditional when rank is 3
  - [] // delete and update

1.4 The seven overloads of ?

[ ]:
```
egAtom:12
egAtom2:6
egList:10+2*til 10
```

[ ]:
```
egList?egAtom / find: returns index of atom if it is in list, else list item count
```

[ ]:
```
egAtom?egList / roll: returns a random k (egAtom) combination of the list with repetition (n can be atom)
```

[ ]:
```
(neg egAtom2)?egList / deal: returns a random k combination of n __without repetition__ (n>=k; n can be atom)
```

[ ]:
```
0N?egList / permute: returns a random permutation of n
```

[ ]:
```
egEnum:`a`b / create enum explicitly
`egEnum?`c / Enum Extend: adds item to enum object
egEnum
```

1.5 String manupulation

- Join return values of system commands with raze

[ ]:
```
(raze system "pwd"),"/my/path/myFile.q"
```

[ ]:
```
.h.ty
```