

# Q LANGUAGE - TYPE CASTING

It is often required to change the data type of some data from one type to another. The standard casting function is the “\$” **dyadic operator**.

Three approaches are used to cast from one type to another *except for string* –

- Specify desired data type by its symbol name
- Specify desired data type by its character
- Specify desired data type by its short value.

## Casting Integers to Floats

In the following example of casting integers to floats, all the three different ways of casting are equivalent –

```
q)a:9 18 27

q)$[`float;a]      / Specify desired data type by its symbol name, 1st way
9 18 27f

q)$["f";a]        / Specify desired data type by its character, 2nd way
9 18 27f

q)$[9h;a]        / Specify desired data type by its short value, 3rd way
9 18 27f
```

Check if all the three operations are equivalent,

```
q)($[`float;a]~$["f";a]) and ($[`float;a] ~ $[9h;a])
1b
```

## Casting Strings to Symbols

Casting string to symbols and vice versa works a bit differently. Let's check it with an example –

```
q)b: ("Hello";"World";"HelloWorld")      / define a list of strings

q)b
"Hello"
"World"
"HelloWorld"

q)c: `$b                                / this is how to cast strings to symbols

q)c
`Hello`World`HelloWorld                / Now c is a list of symbols
```

Attempting to cast strings to symbols using the keyed words `symbol or 11h will fail with the type error –

```
q)b
"Hello"
"World"
"HelloWorld"

q)`symbol$b
'type

q)11h$b
```

'type

## Casting Strings to Non-Symbols

Casting strings to a data type other than symbol is accomplished as follows –

```
q)b:900          / b contain single atomic integer
q)c:string b    / convert this integer atom to string "900"
q)c
"900"
q)`int $ c      / converting string to integer will return the
                  / ASCII equivalent of the character "9", "0" and
                  / "0" to produce the list of integer 57, 48 and
                  / 48.
57 48 48i
q)6h $ c        / Same as above
57 48 48i
q)"i" $ c        / Same as above
57 48 48i
q)"I" $ c
900i
```

So to cast an entire string *thelistofcharacters* to a single atom of data type **x** requires us to specify the upper case letter representing data type **x** as the first argument to the **\$** operator. If you specify the data type of **x** in any other way, it result in the cast being applied to each character of the string.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js