The `wrapr R` package supplies a number of substantial programming tools, including the `S3`/`S4` compatible dot-pipe, `unpack`/`pack` object tools, and many more. It also supplies a number of formatting and parsing convenience tools:

- **`qc()`** ("quoting concatenate"): quotes strings, giving value-oriented interfaces much of the incidental convenience of non-standard evaluation (NSE) interfaces.
- **map_to_char()**: prints maps and vectors as executable code..
- **`let()`**: allows proper value-oriented programming over non-standard evaluation (NSE) interfaces.

I am excited to share one more such convenience interface: `bc()` ("blank concatenate"). `bc()` takes a single string argument, parses it, and builds up a vector of the described values.

`bc()` is easy to demonstrate.

```
library(wrapr)

x <- 1
y <- 2

ls()  # result not easy to paste back into R
# [1] "x" "y"

# the bc() fix
bc('"x" "y"')
# [1] "x" "y"

# the map_to_char() fix
map_to_char(ls())
# [1] "c('x', 'y')"
```

I myself find these solutions a bit more convenient than the usual `dump()`, `deparse()`, `dput()`, `eval()`, or `parse()`.

`bc()` requires the outer quotes, but not the internal quotes. That is: `bc('x y')` and `bc('x,y')` are also equivalent to `c("x", "y")`.

And that is some of the tools that make using `R` results and error messages to progress on projects by producing new `R` code easier. …