action

- **primitive** (provided by eli)
  - **functions**
    - **monadic** (`f A`)
    - **dyadic** (`B f A`)
    - **scalar functions**
      - *arithmetic* `+ - * *. % %. | ~. _. @ n@`
      - *logical* `~ ^ &`
      - *relational* `= < > ~= >= <=`
    - **mixed functions**
      - *(indexed)assignment* `<- A[I]<-B`
      - *indexing* `A[I], M[I;J]`
      - *structural transf.* `# , $ $. ^. !.`
      - *others* `! ? ?. < > <. >. +. &. #.`
  - **operator** (*derived functions*)
    - *reduction* `f/`
    - *scan* `f\`
    - *inner product* `f:g`
    - *outer product* `.:f`
    - *axis operator* `f[n]`

- **user defined function**
  - *monadic, dyadic, niladic*
  - *having result or no result*
  - **form**
    - *standard*
    - *short*

- *code organization*
  - *simple line/branch line* (`->`)
  - *structural statements* — `if else case for while`
  - *functions*
  - *workspace*
  - *script files*
  - *library**

all functions have equal precedence
a (simple) line of code executes from right to left but respects parenthesis
for a dyadic scalar function shapes of operators on both sides must be conformable
   i.   one side is a scalar or one-element
   ii.   both sides have the same shape