0. APL "scalar" functions + - × ÷ ⌊ ⌈ | * ⊛ ○ ... are defined on single values and generalize in a uniform way to arrays of arbitrary shape and depth.

1. We usually assign both a monadic and dyadic function to each of the primitive symbols.

2. APL uses dyadic circle for 25 primitive circle (trig) functions

```
     ...
    ¯2○  arccos
    ¯1○  arcsin
     0○  sqrt(1-n^2)
     1○  sin
     2○  cos
     ...
```

3. A monadic use of Pi-times gives some pleasing expressions:

```
      1 2 3○○÷6 3 4   ⍝ sin(pi/6), cos(pi/3), tan(pi/4)
0.5 0.5 1
```

⍝ Engineer's proof that sin-squared + cos-squared = 1:

```
      +⌿(1 2∘.○○0.1×⍳20)*2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

John