FUNCTIONAL PROGRAMMING IN NYQUIST

Programs are expressions!

Functional Programming

- Program in terms of functions and values
- NOT VARIABLES
- Compose functions: f(g(x), h(x)) to get complex behaviors
- DO NOT MAKE MANY STEPS AND STATE CHANGES TO GET COMPLEX BEHAVIORS

A Very Stateful Program

```
variable sum
function init(x) sum = x
function addx(x) sum += x
function multx(x) sum *= x
function mysound()
  begin
    exec init(hzosc(440.0))
    loop for i from 2 to 10
      exec addx(hzosc(440.0 * i) * rrandom()))
    end
    exec multx(env(0.05, 0.2, 0.5, 1, 0.5, 0.2))
  end
exec mysound()
play sum
```


A Functional Program

```
function rand-harm(hz) return hzosc(hz) * rrandom()

function harmonics(hz, n)
  begin
    if n = 1 then
      return rand-harm(hz)
    else
      return rand-harm(hz * n) + harmonics(hz, n - 1)
  end

function mysound()
    return harmonics(440.0, 10) *
           env(0.05, 0.2, 0.5, 1, 0.5, 0.2)

play mysound()
```


Mostly Functional, Local Variables

```
function harmonics(hz, n)
  begin
    with snd = hzosc(hz * n) * rrandom()
    if n > 1 then
      set snd += harmonics(hz, n - 1)
    return snd
  end

function mysound()
    return harmonics(440.0, 10) *
           env(0.05, 0.2, 0.5, 1, 0.5, 0.2)

play mysound()
```


A Better Functional Program

```
function harmonics(hz, n)
  return simrep(i, n,
                hzosc(hz * (i + 1)) * rrandom())

function mysound()
  return harmonics(440.0, 10) *
         env(0.05, 0.2, 0.5, 1, 0.5, 0.2)

play mysound()
```