

CONTROL CONSTRUCTS

Conditionals, Loops, and More Fun
With Nyquist

If-Then-Else

```
if pitch > C4 then
  return flute(pitch)
else
  return tuba(pitch)
```

```
if velocity > 127 then
  set velocity = 127
```

- **then** and **else** are followed by single command
- Use **begin-end** to contain multiple commands
- Avoid **if c1 then if c2 then s1 else s2**
 - Which **if** does the **else** belong to?
 - Use **begin-end** to disambiguate

Loop Command

- Basic syntax is just `loop commands end`
 - Use `with` to declare local variables
 - `for i from 0 below 10` - $i = 0, 1, \dots, 9$
 - `for elem in my-list` - iterate over list elements
 - `for v = init then update` - flexible update
 - `while expression` - arbitrary stop condition
 - `until expression` - arbitrary stop condition
 - `repeat n` - iterate n times
 - `finally return local-variable` - executed once at end
- Many options! See Ref. Manual. Next up: SAL examples...

Example

```
function pluck-chord(pitch, interval, n)
  begin
    with s = pluck(pitch)
    loop
      for i from 1 below n
        set s += pluck(pitch + interval * i)
      end
    return s
  end

play pluck-chord(c3, 5, 2)
play pluck-chord(d3, 7, 4) ~ 3
play pluck-chord(c2, 10, 7) ~ 8
```