```
var a   declare a variable

{ ← organize code in blocks
   print foo  print to the terminal
}

fn f(x) {  define a function
   return x^2
}      return a value

foo(f)    functions can be passed to other
          functions!

   parent class ↘
class Foo is Boo {  define a class
   init(p) {  ...with methods like this initializer
      self.prop = p
   }       ← set and access object
                    properties
}

a = Foo("hey")  create an object

a.foo()  invoke a method
```

```
if (i<1) ▨ else ▨   conditionally run code

for (a in b) { ▨ }   loop over the contents
                     of a collection

do { ▨ } while (a<b)   conventional loops
                        too
while (a<b) { ▨ }

try { ▨ } catch { ▨ }   deal with
                         anticipated errors

a = [ 1, 2, 3 ]              lists
b = { "a": 1, "b": 2 }       ...and dictionaries

"Hello world ${i}"  strings (with interpolation)

m=Matrix(2,2)   dense and sparse matrices
m[0,0]=1   set and access elements
           of a collection by
           indexing

import optimize    extensible,
                    modular
```