Here is an example SLiM script. It's not the simplest, but it does demonstrate a number of different language features to be syntax highlighted.

Note that below we include labels on particular lines, so we can refer to them. For instance, at line 21 we use the function defined later, at line 88. The way this works is that the characeters `*@` and `@*` delimit code that is "escaped" to LaTeX.

```
initialize() {
  initializeSLiMModelType("nonWF");
  initializeSLiMOptions(dimensionality="xy");

  defaults = Dictionary(
    "SEED", getSeed(),
    "SD", 0.3,          // sigma_D, dispersal distance
    "SX", 0.3,          // sigma_X, interaction distance for
    measuring local density
    "SM", 0.3,          // sigma_M, mate choice distance
    "K", 5,             // carrying capacity per unit area
    "LIFETIME", 4,      // average life span
    "WIDTH", 25.0,      // width of the simulated area
    "HEIGHT", 25.0,     // height of the simulated area
    "RUNTIME", 200,     // total number of ticks to run the
    simulation for
    "L", 1e8,           // genome length
    "R", 1e-8,          // recombination rate
    "MU", 0             // mutation rate
    );

  // Set up parameters with a user-defined function
    setupParams(defaults);

  // Set up constants that depend on externally defined parameters
  defineConstant("FECUN", 1 / LIFETIME);
  defineConstant("RHO", FECUN / ((1 + FECUN) * K));
  defineConstant("PARAMS", defaults);

  setSeed(SEED);

  // basic neutral genetics
  initializeMutationRate(MU);
  initializeMutationType("m1", 0.5, "f", 0.0);
  initializeGenomicElementType("g1", m1, 1.0);
  initializeGenomicElement(g1, 0, L-1);
  initializeRecombinationRate(R);

  // spatial interaction for local density measurement
  initializeInteractionType(1, "xy", reciprocal=T, maxDistance=3 *
    SX);
  i1.setInteractionFunction("n", 1, SX);

  // spatial interaction for mate choice
  initializeInteractionType(2, "xy", reciprocal=T, maxDistance=3 *
    SM);
  i2.setInteractionFunction("n", 1, SM);
}

1 first() {
```

```
  sim.addSubpop("p1", asInteger(K * WIDTH * HEIGHT));
  p1.setSpatialBounds(c(0, 0, WIDTH, HEIGHT));
  p1.individuals.setSpatialPosition(p1.pointUniform(p1.
    individualCount));
}

first() {
  // preparation for the reproduction() callback
  i2.evaluate(p1);
}

reproduction() {
  mate = i2.drawByStrength(individual, 1);
  if (mate.size())
    subpop.addCrossed(individual, mate, count=rpois(1, FECUN));
}

early() {
  // Disperse offspring
  offspring = p1.subsetIndividuals(maxAge=0);
  p1.deviatePositions(offspring, "reprising", INF, "n", SD);

  // Measure local density and use it for density regulation
  i1.evaluate(p1);
  inds = p1.individuals;
  competition = i1.localPopulationDensity(inds);
  inds.fitnessScaling = 1 / (1 + RHO * competition);
}

late() {
  if (p1.individualCount == 0) {
    catn("Population went extinct! Ending the simulation.");
    sim.simulationFinished();
  }
}

RUNTIME late() {
  catn("End of simulation (run time reached)");
  sim.treeSeqOutput(OUTPATH, metadata=PARAMS);
  sim.simulationFinished();
}

function (void)setupParams(object<Dictionary> defaults)
{
  if (!exists("PARAMFILE")) defineConstant("PARAMFILE", "./params.
    json");
  if (!exists("OUTDIR")) defineConstant("OUTDIR", ".");
  defaults.addKeysAndValuesFrom(Dictionary("PARAMFILE", PARAMFILE,
    "OUTDIR", OUTDIR));

  if (fileExists(PARAMFILE)) {
    defaults.addKeysAndValuesFrom(Dictionary(readFile(PARAMFILE)));
    defaults.setValue("READ_FROM_PARAMFILE", PARAMFILE);
  }

  defaults.setValue("OUTBASE", OUTDIR + "/out_" + defaults.getValue
    ("SEED"));
```

```
100   defaults.setValue("OUTPATH", defaults.getValue("OUTBASE") + ".
        trees");
101
102   for (k in defaults.allKeys) {
103     if (!exists(k))
104       defineConstant(k, defaults.getValue(k));
105     else
106       defaults.setValue(k, executeLambda(k + ";"));
107   }
108
109   // print out default values
110   catn("============================");
111   catn("Model constants: " + defaults.serialize("pretty"));
112   catn("============================");
113 }
```