For defining Sequences and Regimens, we have decided to use YAML.

References to sub-sequences can be done with their names.

The ID of regimens and sequences, which is defined by the FarmBot when it receives it, will be written to file automatically.

Examples of Sequences and Regimens:
YAML to define sequences with or without loops:

Example of a YAML file:
"""
CSV: my_map.csv
other_files: ["yaml_file1", "yaml_file2"]  # This field is optional, the program will always look in its own file first.

name_set_by_user_1:
 kind: "sequence"
 ID : <<created and set automatically by the app>>
 color : "gray"
 actions: [TO_PLANT: "my_favorite_plant", WAIT: "1000", MOVE_ABS: {x: 0, y: 0, z: 0, x_off: 0, y_off: 0, z_off: 0, speed: 100}]

name_set_by_user_2:
 kind: "sequence"
 ID : <<created and set automatically by the app>>
 color: "gray"
 group: ["11", "funny_name"] # iterate over all members of this group
 type: ["broccoli", "onion"] # iterate over all members of this type
 actions: [TO_SELF, SEQ: "water_stuff_sequence"]

YAML Regimens:
name_set_by_user_3:
 kind: "regimen"
 ID : <<created and set automatically by the app>>
 color: "green"
 schedule: [{name: "sequence_name1", days: [1,2,5], time: "23:00"},
          {name: "sequence_name2", days: [11,12,15], time: "21:00"}]
"""

The following are the keywords and formats we defined for single FarmBot commands such as Move_Absolute and Write_Pin.

 MOVE_ABS: {x: 0, y: 0, z: 0, x_off: 0, y_off: 0, z_off: 0, speed: 100}
 MOVE_REL: {x: 0, y: 0, z: 0, speed: 100}
 IF : {cond: <<see below>>, then: , else: }

 cond: "x = 0 AND y > 0 OR PIN_1 < 0 OR PIN_A14 = UNKNOWN"

 FIND_HOME : [x,y,z]
 MESSAGE : {text: <<note: {{x/y/z}} is a special tag>>, type: "success/warning/busy/error/info"}
 WAIT : "time in milliseconds"

READ_PIN : {pin : "<<number>>", label: "", mode : "D/A"}
WRITE_PIN : {pin : "<<number>>", value: "ON/OFF", mode : "D/A"}

TO_SELF # Go to the current plant in a loop
TO_PLANT  # go to a plant with this (unique) name, this saves typing in coordinates by hand if you have a special plant.
SEQ # get a sub-sequence