```
# What to do with a YAML file

# Someone else write

get_id_back(CeleryScript, UUID)

delete_object("regimen", "id")

unique_name() # returns a name not currently used in internal storage

# Rachel writes these:

load_commands(yaml_file):
  self.CSV = yaml_file{"CSV"}
  for yaml_object in yaml_file:
    make_yaml_object(yaml_object)

make_yaml_object(yaml_object):
  if "start_time" in yaml_object:
    make_farm_event(yaml_object)
  else if "schedule" in yaml_object:
    make_regimen(yaml_object)
  else if "actions" in yaml_object:
    make_sequence(yaml_object)
  else:
    "Error: YAML object is not correctly formatted."

make_farm_event(yaml_object):
  write the start of the CeleryScript
  if "repeat_event" in yaml_object:
    write parts of the CeleryScript a little differently so the event repeats (DONE)
  if "schedule" in yaml_object:
    take the "schedule" part of the YAML object and send it to make_regimen()
    use the ID make_regimen() returns to finish writing the CeleryScript
  else if "actions" in yaml_object:
    if type(item["actions"]) is not str: # if we need to make a sequence:
      take the "actions" part of the YAML object and send it to make_sequence()
```

```
    else: # if the "actions" refer to the name of a sequence defined somewhere else in the file
      get the id and type of the object by calling obj_from_name()
    use the ID make_sequence() returns to finish writing the CeleryScript
  Send the CeleryScript and get back the ID
  Write a Farm Event YAML object to internal storage in the corrent format, with the ID
  return

make_regimen(yaml_object):
  if "name" not in yaml_object:
    auto = True
  else:
    auto = False
    check if the regimen is already in the internal storage, if it is, we have to update and delete a lot
of things (for later)
  write the start of the CeleryScript
  for item in yaml_object["schedule"]:
    if type(item["actions"]) is not str: # if we need to make a sequence
      send the "actions", "groups", and "types" as a single YAML object to make_sequence()
    else: # if the "actions" refer to the name of a sequence defined somewhere else in the file
      find the sequence in the file and send it to make_sequence()
    take the returned ID of the sequence and finish writing your CeleryScript
  Send the CeleryScript and get back the ID
  Write a Regimen YAML object to internal storage in the corrent format, with the ID
  return Regimen_ID

  make_sequence(yaml_object):
    if "name" not in yaml_object:
      auto = True
    else:
      auto = False
      check if the sequence is already in the internal storage, if it is, we have to update and delete a
lot of things (for later)
    write the start of the CeleryScript
    if "group" in yaml_object:
      loop over the entire CSV:
        if row is the right "group":
          CeleryScript + make_actions(yaml_object["actions"], x, y, z)
```

```
    else if "group" in yaml_object:
      loop over the entire CSV:
        if row is the right "group":
        CeleryScript + make_actions(yaml_object["actions"], x, y, z)
    else if "type" in yaml_object:
      loop over the entire CSV:
        if row is the right "type":
        CeleryScript + make_actions(yaml_object["actions"], x, y, z)
    Send the CeleryScript and get back the ID
    Write a Sequence YAML object to internal storage in the corrent format, with the ID
    return Sequence_ID

name_1:
  kind: "farm_event"
  auto: False
  ID: #

name_2:
  kind: "regimen"
  auto: True/False
  ID: #

name_3:
  kind: "sequence"
  auto: True/False
  ID: #
```