Sprinkle Irrigation Controller

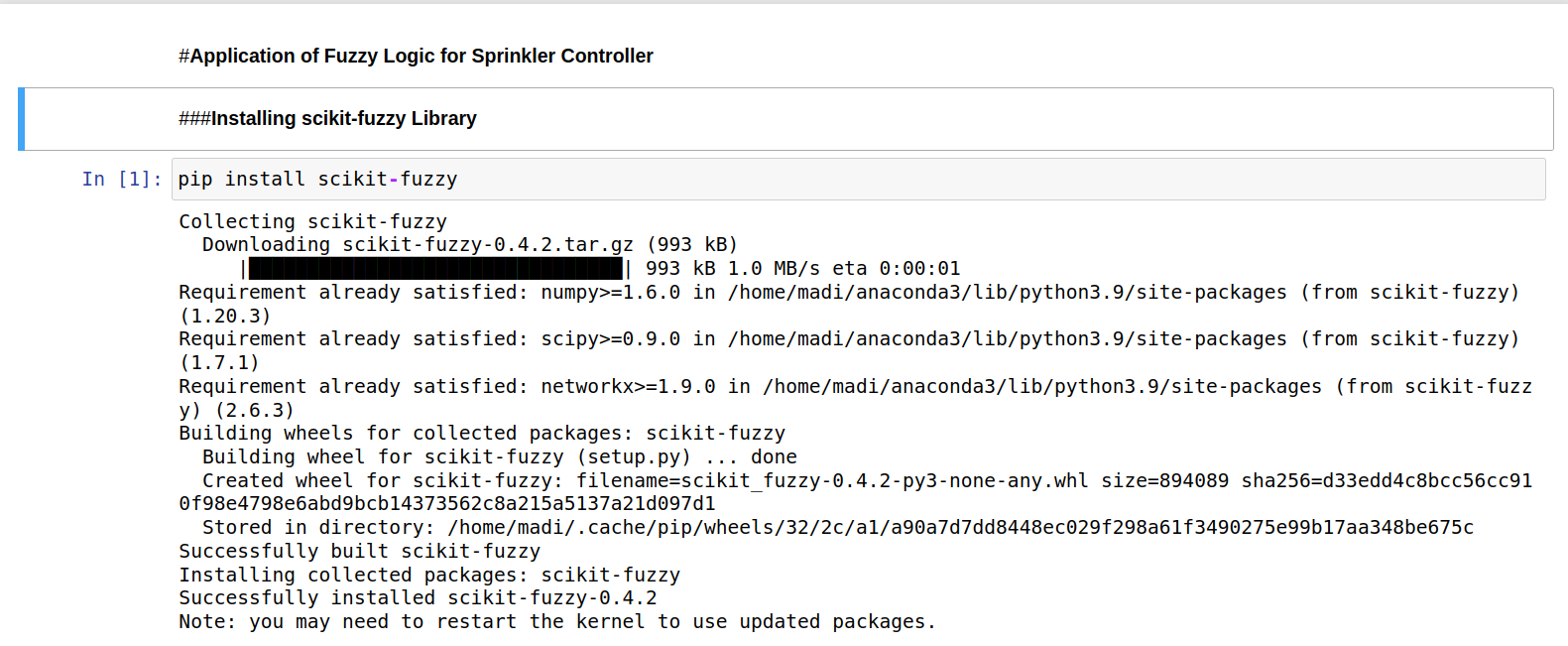
* Application of Fuzzy Logic for Sprinkler Controller in Drip Irrigation.
* Controlling how long a sprinkler should function given the current Temperature and Humidity.
* Inputs:

1. Temperature: In Degree Celsius.
2. Humidity: In Percentage(%).

* Output:

Sprinkler Duration: In Minutes.

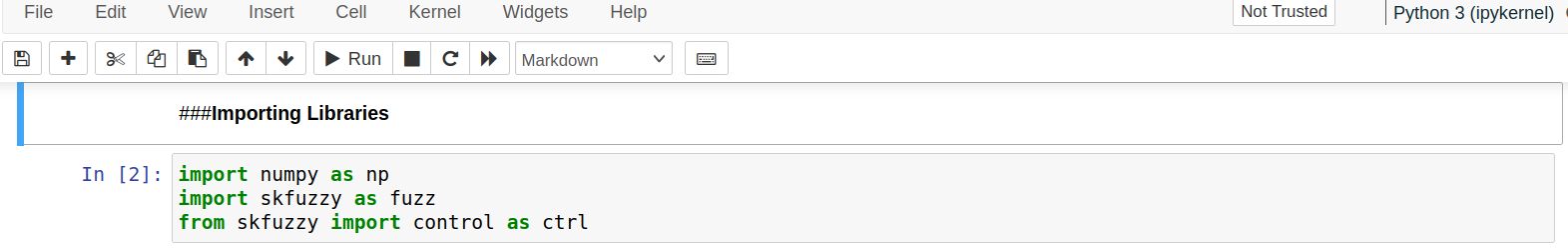
* Implementation of code for the application…
* pip install scikit-fuzzy.



* import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl



* HUMIDITY = 'humidity'

SPRINKLER\_DURATION = 'sprinkler\_duration'

TEMPRATURE = 'temperature'

# Temperature's fuzzy linguistics

COLD = 'Cold'

COOL = 'Cool'

NORMAL = 'Normal'

WARM = 'Warm'

HOT = 'Hot'

# Humidity's fuzzy linguistics

DRY = 'Dry'

MOIST = 'Moist'

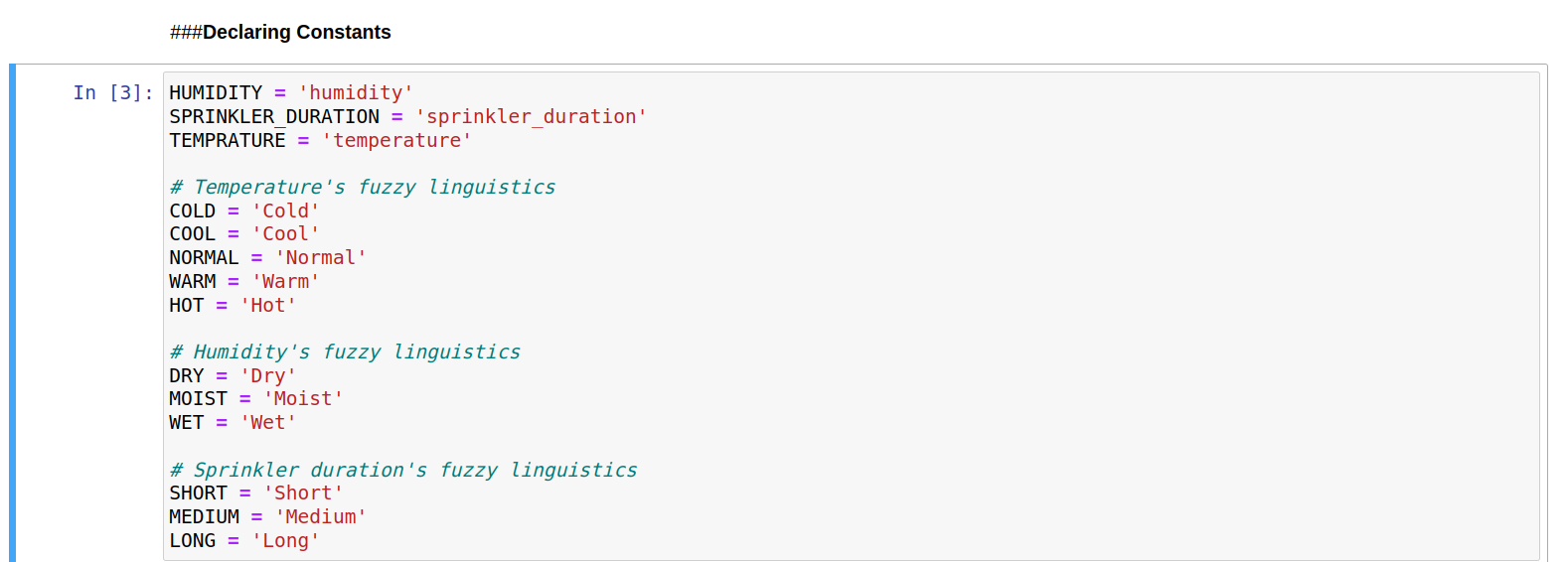
WET = 'Wet'

# Sprinkler duration's fuzzy linguistics

SHORT = 'Short'

MEDIUM = 'Medium'

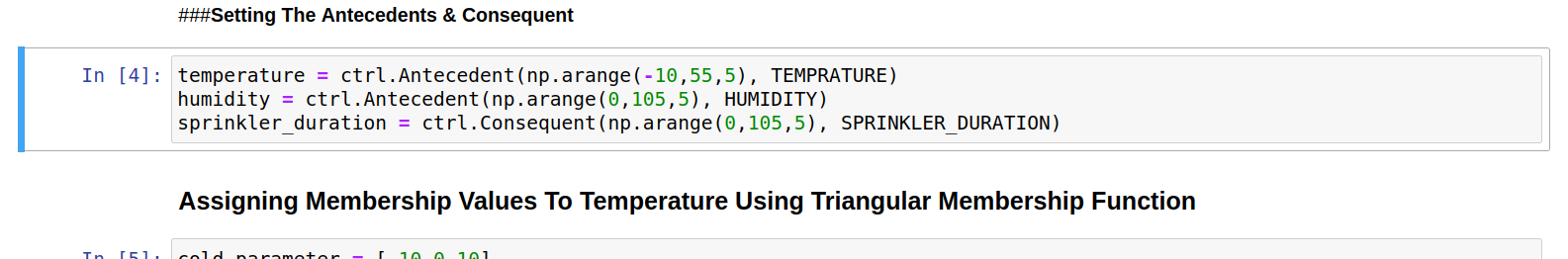
LONG = 'Long'



* temperature = ctrl.Antecedent(np.arange(-10,55,5), TEMPRATURE)

humidity = ctrl.Antecedent(np.arange(0,105,5), HUMIDITY)

sprinkler\_duration = ctrl.Consequent(np.arange(0,105,5), SPRINKLER\_DURATION)



* cold\_parameter = [-10,0,10]

cool\_parameter = [0,10,20]

normal\_parameter = [10,20,30]

warm\_parameter = [20,30,40]

hot\_parameter = [30,40,50]

temperature[COLD] = fuzz.trimf(temperature.universe, cold\_parameter)

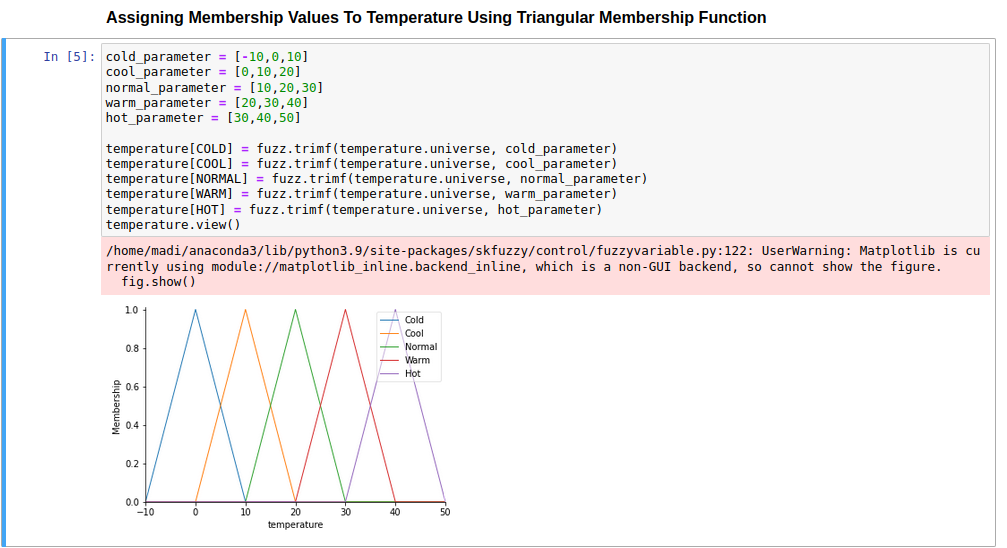
temperature[COOL] = fuzz.trimf(temperature.universe, cool\_parameter)

temperature[NORMAL] = fuzz.trimf(temperature.universe, normal\_parameter)

temperature[WARM] = fuzz.trimf(temperature.universe, warm\_parameter)

temperature[HOT] = fuzz.trimf(temperature.universe, hot\_parameter)

temperature.view()



* dry\_parameter = [0,25,50]

moist\_parameter = [25,50,75]

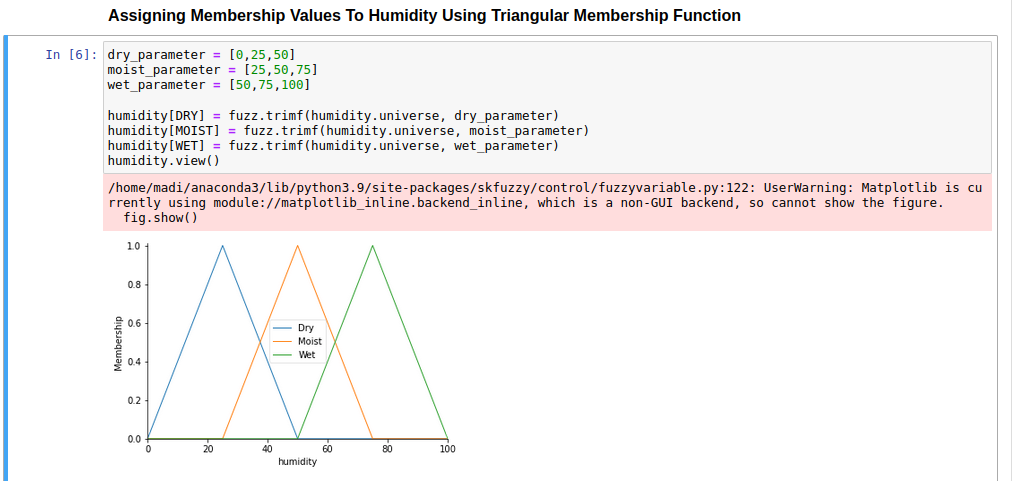
wet\_parameter = [50,75,100]

humidity[DRY] = fuzz.trimf(humidity.universe, dry\_parameter)

humidity[MOIST] = fuzz.trimf(humidity.universe, moist\_parameter)

humidity[WET] = fuzz.trimf(humidity.universe, wet\_parameter)

humidity.view()



* short\_parameter = [0,25,50]

medium\_parameter = [25,50,75]

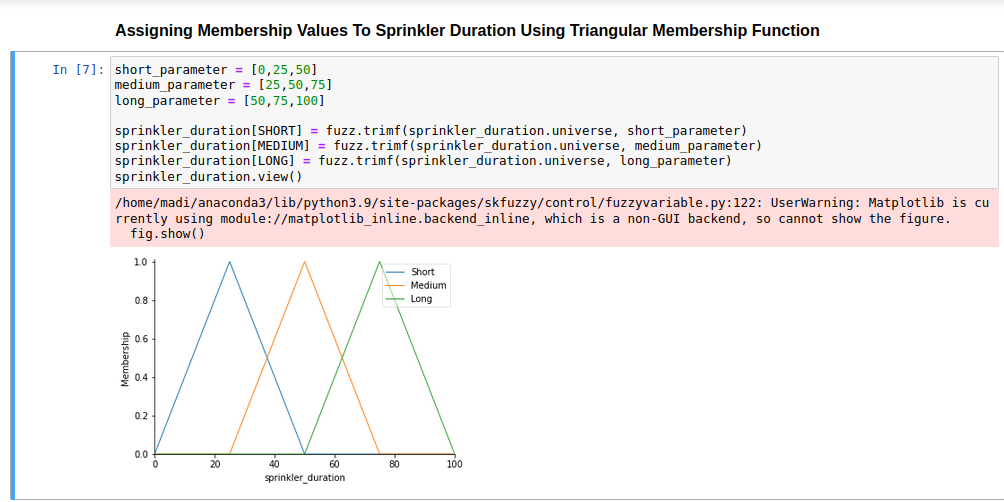
long\_parameter = [50,75,100]

sprinkler\_duration[SHORT] = fuzz.trimf(sprinkler\_duration.universe, short\_parameter)

prinkler\_duration[MEDIUM] = fuzz.trimf(sprinkler\_duration.universe, medium\_parameter)

sprinkler\_duration[LONG] = fuzz.trimf(sprinkler\_duration.universe, long\_parameter)

sprinkler\_duration.view()



* rule1 = ctrl.Rule(humidity[DRY] & temperature[COLD], sprinkler\_duration[SHORT])

rule2 = ctrl.Rule(humidity[DRY] & temperature[COOL], sprinkler\_duration[SHORT])

rule3 = ctrl.Rule(humidity[DRY] & temperature[NORMAL], sprinkler\_duration[MEDIUM])

rule4 = ctrl.Rule(humidity[DRY] & temperature[WARM], sprinkler\_duration[LONG])

rule5 = ctrl.Rule(humidity[DRY] & temperature[HOT], sprinkler\_duration[LONG])

rule6 = ctrl.Rule(humidity[MOIST] & temperature[COLD], sprinkler\_duration[SHORT])

rule7 = ctrl.Rule(humidity[MOIST] & temperature[COOL], sprinkler\_duration[SHORT])

rule8 = ctrl.Rule(humidity[MOIST] & temperature[NORMAL], sprinkler\_duration[MEDIUM])

rule9 = ctrl.Rule(humidity[MOIST] & temperature[WARM], sprinkler\_duration[MEDIUM])

rule10 = ctrl.Rule(humidity[MOIST] & temperature[HOT], sprinkler\_duration[LONG])

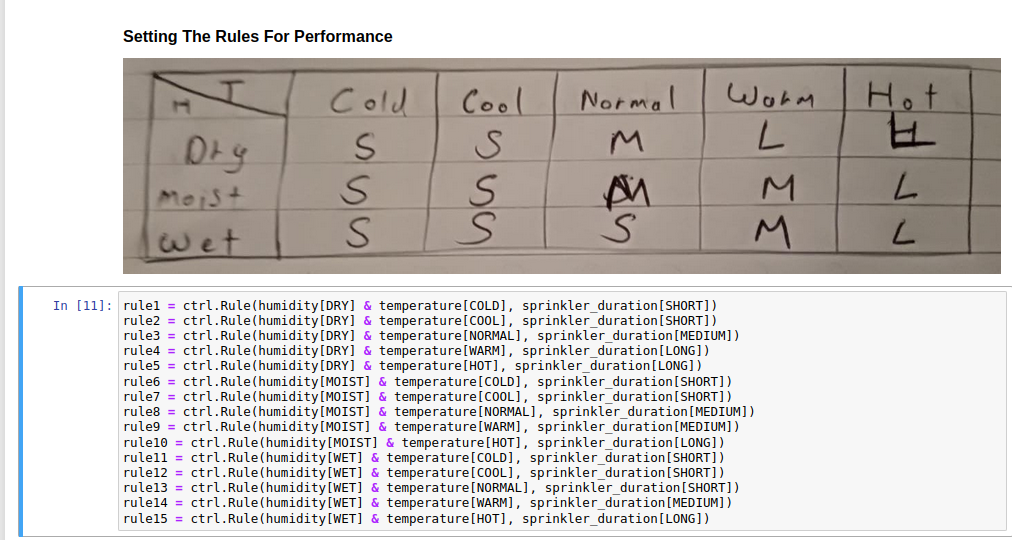
rule11 = ctrl.Rule(humidity[WET] & temperature[COLD], sprinkler\_duration[SHORT])

rule12 = ctrl.Rule(humidity[WET] & temperature[COOL], sprinkler\_duration[SHORT])

rule13 = ctrl.Rule(humidity[WET] & temperature[NORMAL], sprinkler\_duration[SHORT])

rule14 = ctrl.Rule(humidity[WET] & temperature[WARM], sprinkler\_duration[MEDIUM])

rule15 = ctrl.Rule(humidity[WET] & temperature[HOT], sprinkler\_duration[LONG])



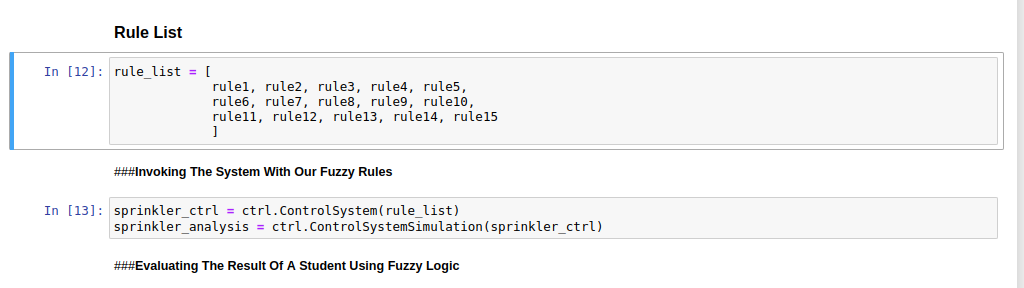
* rule\_list = [

rule1, rule2, rule3, rule4, rule5,

rule6, rule7, rule8, rule9, rule10,

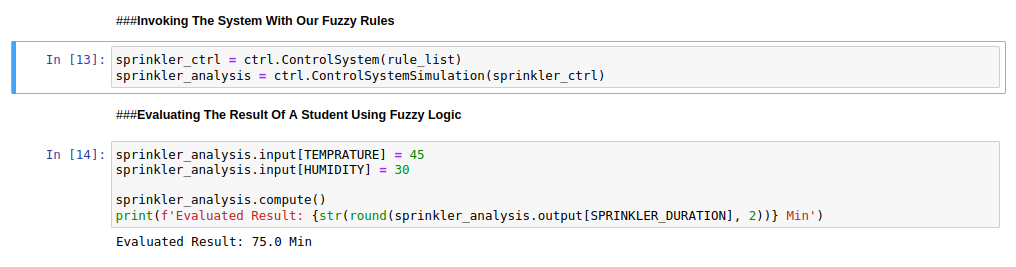
rule11, rule12, rule13, rule14, rule15

]



* sprinkler\_ctrl = ctrl.ControlSystem(rule\_list)

sprinkler\_analysis = ctrl.ControlSystemSimulation(sprinkler\_ctrl)

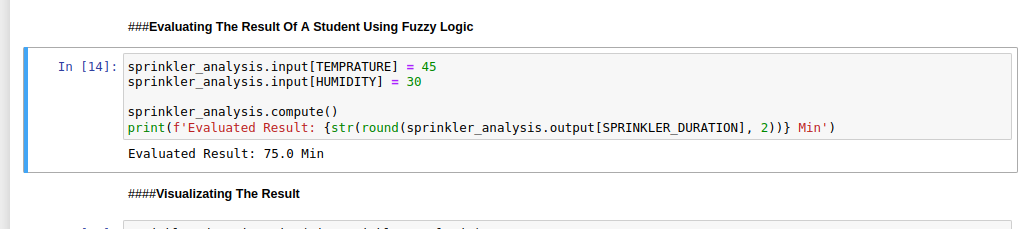


* sprinkler\_analysis.input[TEMPRATURE] = 45

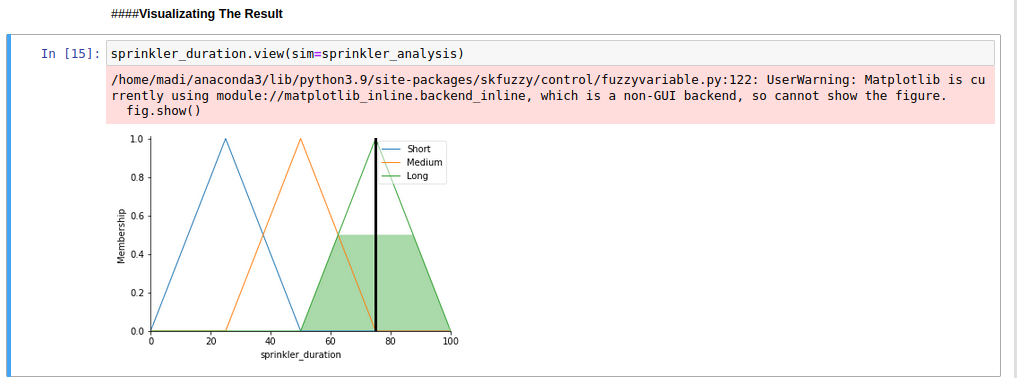
sprinkler\_analysis.input[HUMIDITY] = 30

sprinkler\_analysis.compute()

print(f'Evaluated Result: {str(round(sprinkler\_analysis.output[SPRINKLER\_DURATION], 2))} Min')



* sprinkler\_duration.view(sim=sprinkler\_analysis)



* Reference : <https://github.com/Ashley-J-George/Sprinkle_Irrigation_Controller>
* Name : Ramah Hashem madi.