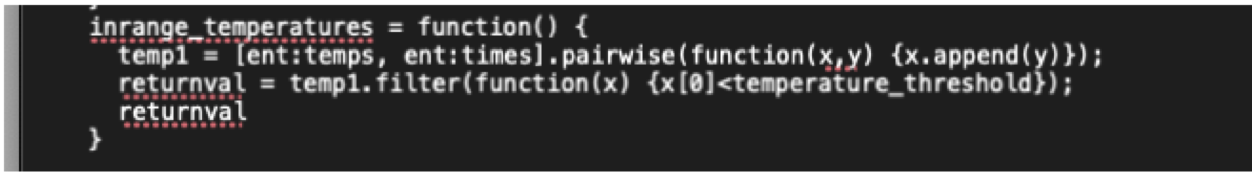
Trav Feller
Lab 3

1. Explain how the collect_temperatures rule and the temperatures function work as an event-query API.
    - Once the collect_temperatures rule has been fired via the client calling the event, it updates the entity variables that store the temps and the timestamps. This does not send anything back to the client. Now if the client wants to see the changes, they have to call the temperatures function via a query to get the data that was updated, in our case, the temps an timestamps.
2. Explain your strategy for finding temperatures that are in range.
    - 

```
inrange_temperatures = function() {
  temp1 = [ent:temps, ent:times].pairwise(function(x,y) {x.append(y)});
  returnval = temp1.filter(function(x) {x[0]<temperature_threshold});
  returnval
}
```

    - Here, I first get the list of all temps and timestamps grouped together by instance. Then I run the filter() function on that list and I filter out all elements where the 0th element in that element (the 0th element is the temperature in my case) is less than the temperature threshold. This way, an array of only temperature/timestamps that are less than my threshold are returned.
3. What happens if provides doesn't list the name of the temperatures function?
    - “the "**provides**" keyword allows other rulesets that use this one as a module to call the function” (from the documentation). So, if a function is not included in the provides section, we are unable to use this function outside of the ruleset it is created in.
4. What happens if shares doesn't list the name of the temperatures function?
    - If the function is not included in the shares section, the function will not be able to be queried by the API or in the testing tab of the pico engine.