

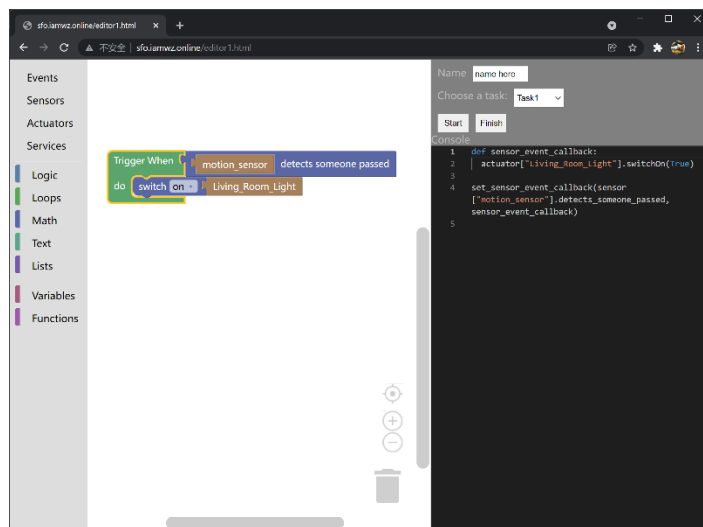
## Holon based Visual Programming Language Tutorial

This software is for non-professional people to design smart home automation by simple drag and drop operations. The picture below is the visual editor.

The area on the left is the editor's toolbox, where you can find the blocks you need. The middle part we called workspace; you can drag your blocks from the toolbox to the workspace to build your workflow by combining them.

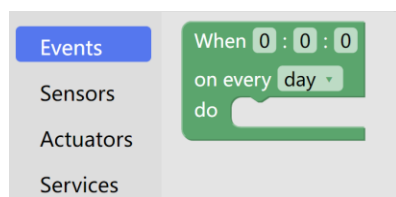
The right part is called code editor; you can see the corresponding Python code of your automation in real-time. In the top right corner, you can enter your name and choose the task you want to try.

When you are ready to start your task, please click the start button, and when you finish, click the finish button.



We simplified all smart home automation elements into four types of blocks

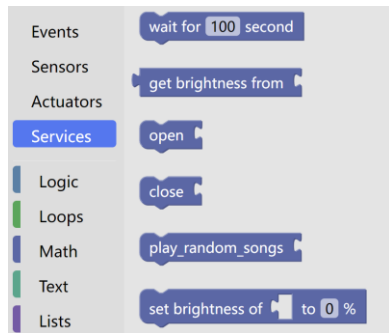
### Event:



All automation should be triggered by an event. You should always choose an appropriate event for your automation. For example, scenes, sensor events and time triggers are the most common events.

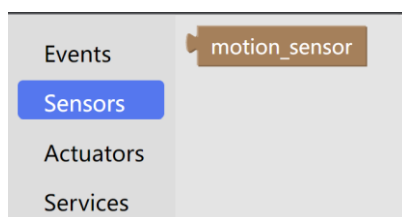
### Service

This section contains all services provided by your smart home devices and other actions like wait (which will let the workflow pause for a designated period)



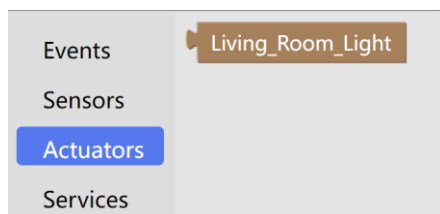
## Sensors

Sensors represent the various types of sensors and switches in a smart home system, which provide services such as triggering events and providing sensor values.



Actuators:

Actuators represent the parts of a smart home system that can be controlled, e.g., lamps, curtains, and other devices that can be controlled to do some actions are included in the actuator category.



## Other sections in the toolbox

Other sections in the toolbox are general programming elements, logic, and loop provide basic control flow of your program. Math can enable you to do calculations or enter a number.

## Configurable Blocks

Some of the blocks are configurable; you can click the gear icon on the left top corner to config a block



Variables and functions provide a higher level of abstraction for your programs.

Sample:

The following sample shows how to design a midnight scene, which will check if the brightness value from the brightness sensor is higher than 200. If so, it will set the brightness of the lamp to 50%. The automation uses a sensor, an actuator, two services and basic if and compare statements from the logic section and number block from the math section.

