Lazy TECH Geek

https://github.com/LazyTechGeek/HomeAssistant-Templates

Templates

Templates in Home Assistant allow you to create **dynamic entities.** They **derive values** from other data using **conditions and logic.** You can **specify templates** for entity properties like names and states.

Supported Template Entities in Home Assistant

(BEFORE 2021.8.1 release)

Entity Type	Key After - platform: template	Purpose
Sensor	**sensors:**	Creates a virtual sensor based on a template value (e.g., temperature, status, power).
Binary Sensor	**binary_sensors:**	A sensor with only on/off states (e.g., motion detected, door open).
Switch	**switches:**	A virtual switch to control other entities or trigger automations.
Light	**lights:**	Creates a light entity that can turn on/off and sometimes adjust brightness.
Fan	**fans:**	Defines a fan entity that can be toggled on/off or adjusted in speed modes.
Cover	**covers:**	Used for garage doors, blinds, shutters, or anything that "opens" and "closes".
Lock	**locks:**	Creates a virtual lock that can be locked/unlocked via automation.
Button	**buttons:**	A stateless button entity that triggers an action when pressed (e.g., a "reset" button).
Number	**numbers:**	A numeric input that can be set within a defined range (e.g., volume control).
Select	**selects:**	A dropdown-style selector for choosing from multiple options (e.g., fan speed mode).

Supported Template Entities in Home Assistant

(AFTER 2021.8.1 release)

Entity Type	Key After template:	Purpose
Sensor	**- sensor:**	Creates a virtual sensor based on a template value (e.g., temperature, status, power).
Binary Sensor	**- binary_sensor:**	A sensor with only on/off states (e.g., motion detected, door open).
Switch	**- switch:**	A virtual switch to control other entities or trigger automations.
Light	**- light:**	Creates a light entity that can turn on/off and sometimes adjust brightness.
Fan	**- fan:**	Defines a fan entity that can be toggled on/off or adjusted in speed modes.
Cover	**- cover:**	Used for garage doors, blinds, shutters, or anything that "opens" and "closes".
Lock	**- lock:**	Creates a virtual lock that can be locked/unlocked via automation.
Button	**- button:**	A stateless button entity that triggers an action when pressed (e.g., a "reset" button).
Number	**- number:**	A numeric input that can be set within a defined range (e.g., volume control).
Select	**- select:**	A dropdown-style selector for choosing from multiple options (e.g., fan speed mode).

Example Comparison

Before 2021.8.1 Release

```
sensor:
  - platform: template
    sensors:
      room_temperature:
        friendly_name: "Room Temperature"
        unit_of_measurement: "°C"
        value_template: "{{ states('sensor.living_room_temperature') | float }}"
```

After 2021.8.1 Release

```
template:
  - sensor:
      - name: "Room Temperature"
        unique_id: room_temperature
        unit_of_measurement: "°C"
        state: "{{ states('sensor.living_room_temperature') | float }}"
```

Jinja Template Syntax Breakdown

Basic syntax:

{{ … }} → Output data

{% … %} → Logic (if statements, loops)

{# … #} → Comments

Core Data Types in Home Assistant Templates

- **String** – Text-based values (e.g., "Living Room Light")
- **Integer** – Whole numbers (e.g., 42, 0, -5)
- **Float** – Decimal numbers (e.g., 3.14, 99.99)
- **Boolean** – True/False values (e.g., True, False)
- **None / Null** – Represents an empty or unknown value (None)

Core Data Types in Home Assistant Templates: Code Examples

<table>
<tr><td>String – Text-Based Values
{{ "Living Room Light" }}</td><td>Output
Living Room Light</td></tr>
<tr><td>Integer – Whole Numbers
{{ 42 + 8 }}</td><td>Output
50</td></tr>
<tr><td>Float – Decimal Numbers
{{ 3.14 * 2 }}</td><td>Output
6.28</td></tr>
<tr><td>Boolean – True/False Values
{{ is_state("light.living_room", "on") }}</td><td>Output
True</td></tr>
<tr><td>None / Null – Empty or Unknown Values
{{ states("sensor.unknown_sensor") }}</td><td>Output
None</td></tr>
<tr><td>Bonus: Handling none Values Safely
{{ states("sensor.temperature") | float(default=0) }}</td><td>Output
23.5 (if unavailable or unknown → defaults to 0.0)</td></tr>
</table>