# Model-Based Reflex Agent Documentation

## 1. Introduction

This document describes the implementation of a Model-Based Reflex Agent that manages a heater system efficiently by remembering its last action and avoiding unnecessary toggling.

## 2. Code Explanation

The ModelBasedReflexAgent class is designed to control a heating system based on temperature inputs. It prevents unnecessary toggling of the heater by storing its last action.

## 3. Class: ModelBasedReflexAgent

Attributes:  
- threshold: The temperature at which the heater should turn on or off.  
- previous\_action: Stores the last action taken (HEATER\_ON or HEATER\_OFF) to avoid redundant changes.

Methods:  
- \_\_init\_\_(self, threshold=22): Initializes the agent with a desired temperature threshold.  
- perceive\_and\_act(self, temperature): Determines whether to turn the heater ON, OFF, or do NOTHING.

## 4. Code Implementation

class ModelBasedReflexAgent:  
 def \_\_init\_\_(self, threshold=22):  
 self.threshold = threshold # Desired temperature  
 self.previous\_action = None # Stores the last action performed  
   
 def perceive\_and\_act(self, temperature):  
 if temperature < self.threshold and self.previous\_action != "HEATER\_ON":  
 action = "HEATER\_ON"  
 elif temperature >= self.threshold and self.previous\_action != "HEATER\_OFF":  
 action = "HEATER\_OFF"  
 else:  
 action = "NO\_CHANGE" # Avoid unnecessary toggling  
   
 self.previous\_action = action if action != "NO\_CHANGE" else self.previous\_action  
 return action  
  
# Example usage  
agent = ModelBasedReflexAgent(threshold=22)  
temperatures = [20, 21, 22, 23, 21, 20, 22, 23]  
  
for temp in temperatures:  
 action = agent.perceive\_and\_act(temp)  
 print(f"Temperature: {temp} -> Action: {action}")