State:

**Other name (if any)**

The State Pattern is also known as Objects for States.

**What it does**

A State Pattern says that "the class behavior changes based on its state". In State Pattern, we create objects which represent various states and a context object whose behavior varies as its state object changes.

**Where to use**

1. When the behavior of object depends on its state and it must be able to change its behavior at runtime according to the new state.  
2. It is used in vendor machine. Because vendor machine has different states like idle, insert coin, select product, cancel etc.  
3. It is used in TV. Because TV has different states like turn on, change channel, change volume, turn off.

**Steps**

1. Context Concrete Class: The Context is the class that contains the object whose behavior changes based on its internal state. It has two functions, one is setState and doAction/handler/request.  
2. State Interface: The State interface or base class defines a common interface for all concrete state classes. This interface typically declares methods that represent the state-specific behavior that the Context can exhibit. It has only one function which will execute the task, the function is doAction/handler/request.  
3. Concrete State Class: These classes implement the function of the state interface.  
4. Client Code: Create object of context concrete class using context concrete class. Then pass concrete state class using the setState function of the context concrete class.

**Special cases (if any)**

placeholder

**Advantages**

1. It makes any state transitions explicit.

**Disadvantages**

placeholder

**Code**

Geekforgeeks

**Difference with similar pattern**

placeholder

**Diagram**

Tutorialspoint + Geekforgeeks