

# Managing Vue Application State with Pinia



**Jim Cooper**

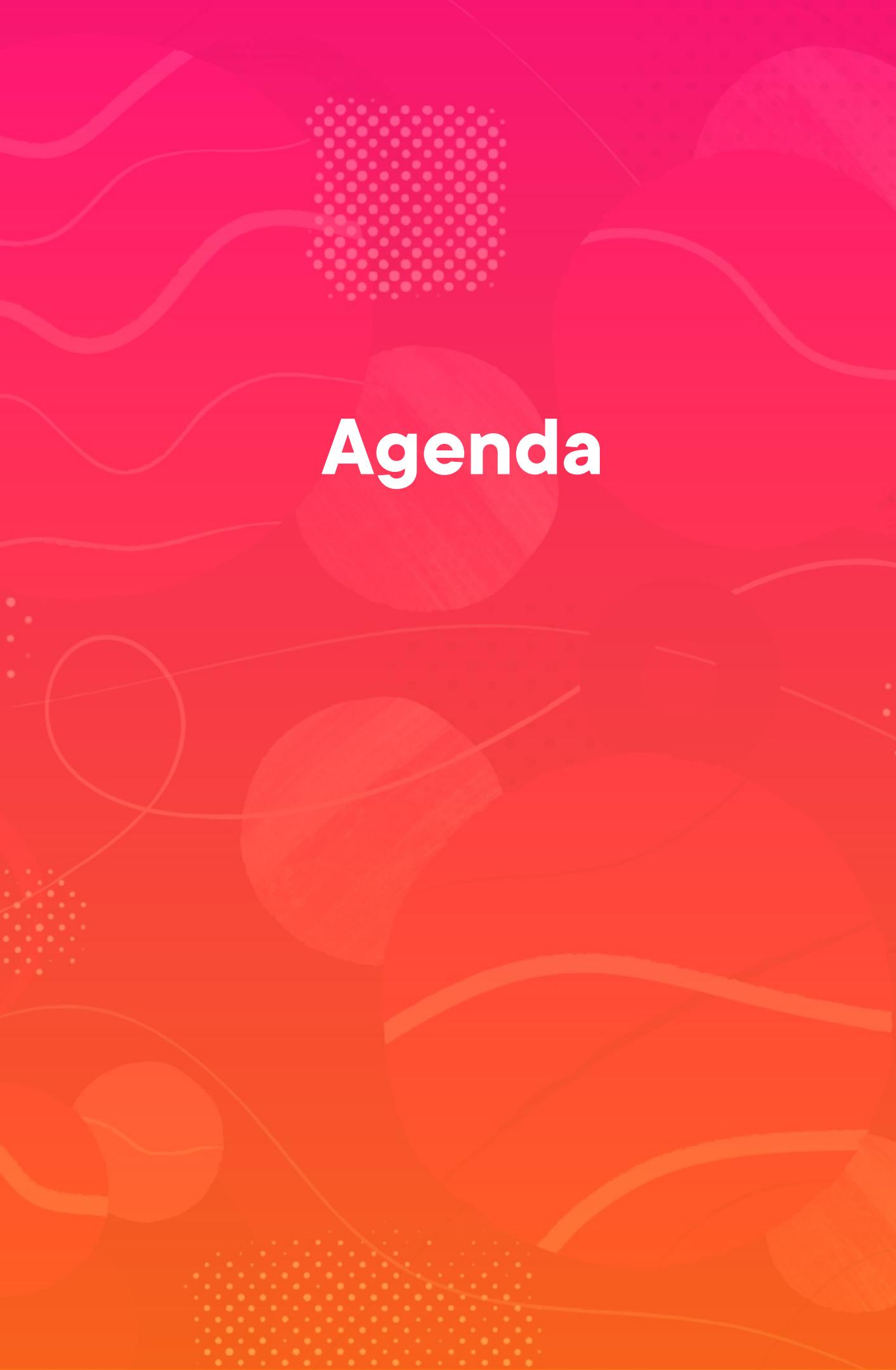
Software Engineer

@jimthecoop | jcoop.io



Pinia





# Agenda

- What is state management?
- Creating a Pinia store
- Storing and accessing data in a Pinia store
- Managing reactivity with store state
- Using calculated data (getters)
- Using actions to fetch data from an API
- Setup stores vs Options stores
- Accessing stores from Composition/Options API components



# What is State Management?

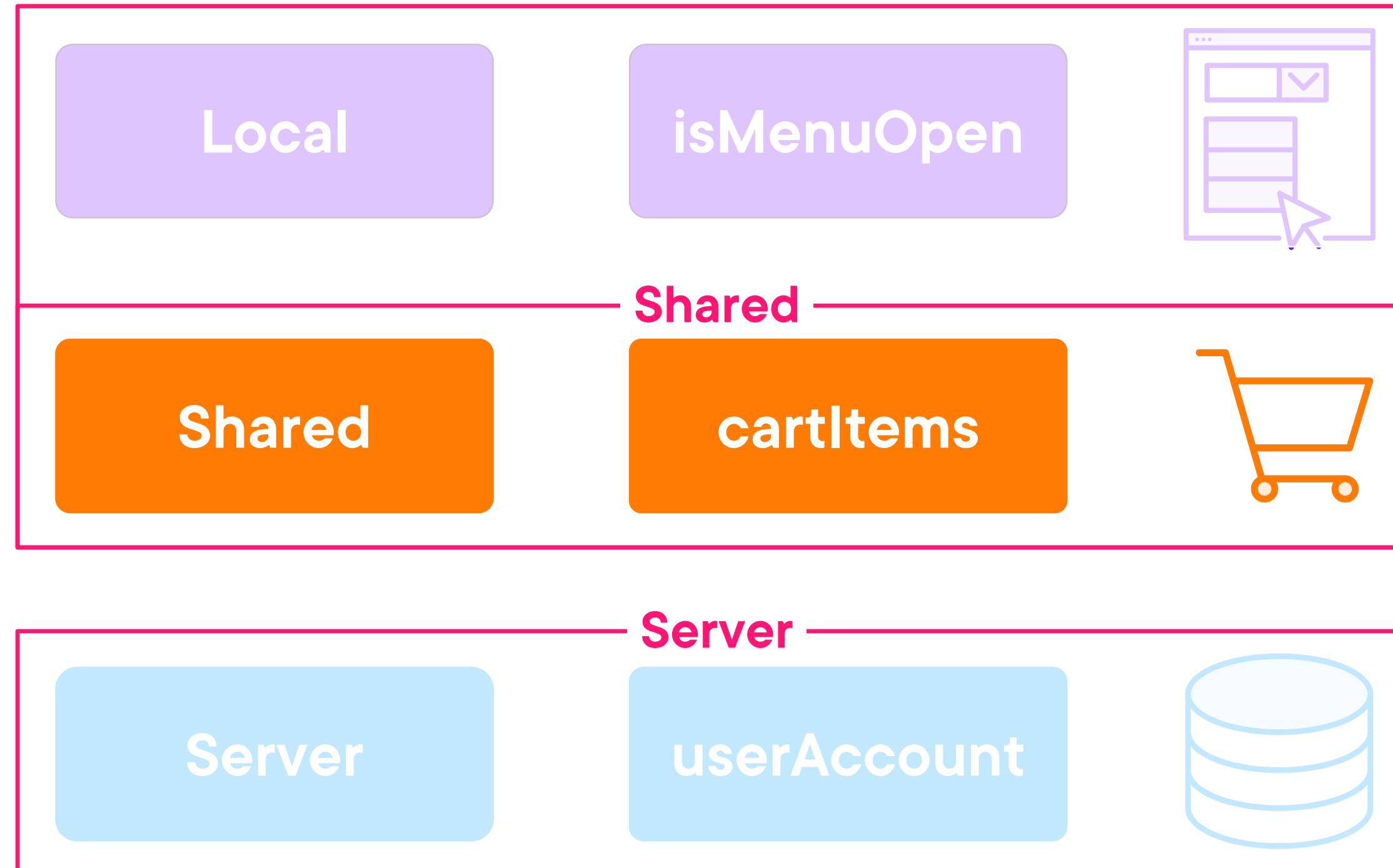


# State

**Data** that describes the **current state** of your application.

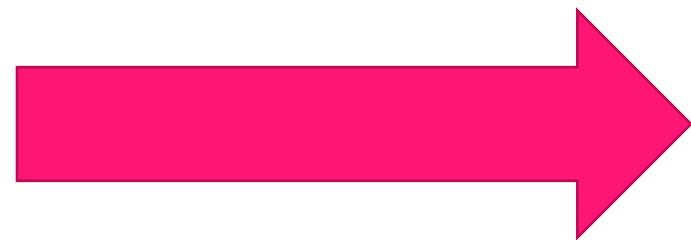


# Different Types of State



# Vuex vs. Pinia

Vuex



Pinia



# Pinia Store Anatomy



# Options Store

State

Data

Actions

Methods

Getters

Computed

# Setup Store



# Options Store

State

Actions

Getters

# Setup Store

Properties

Methods

Computed



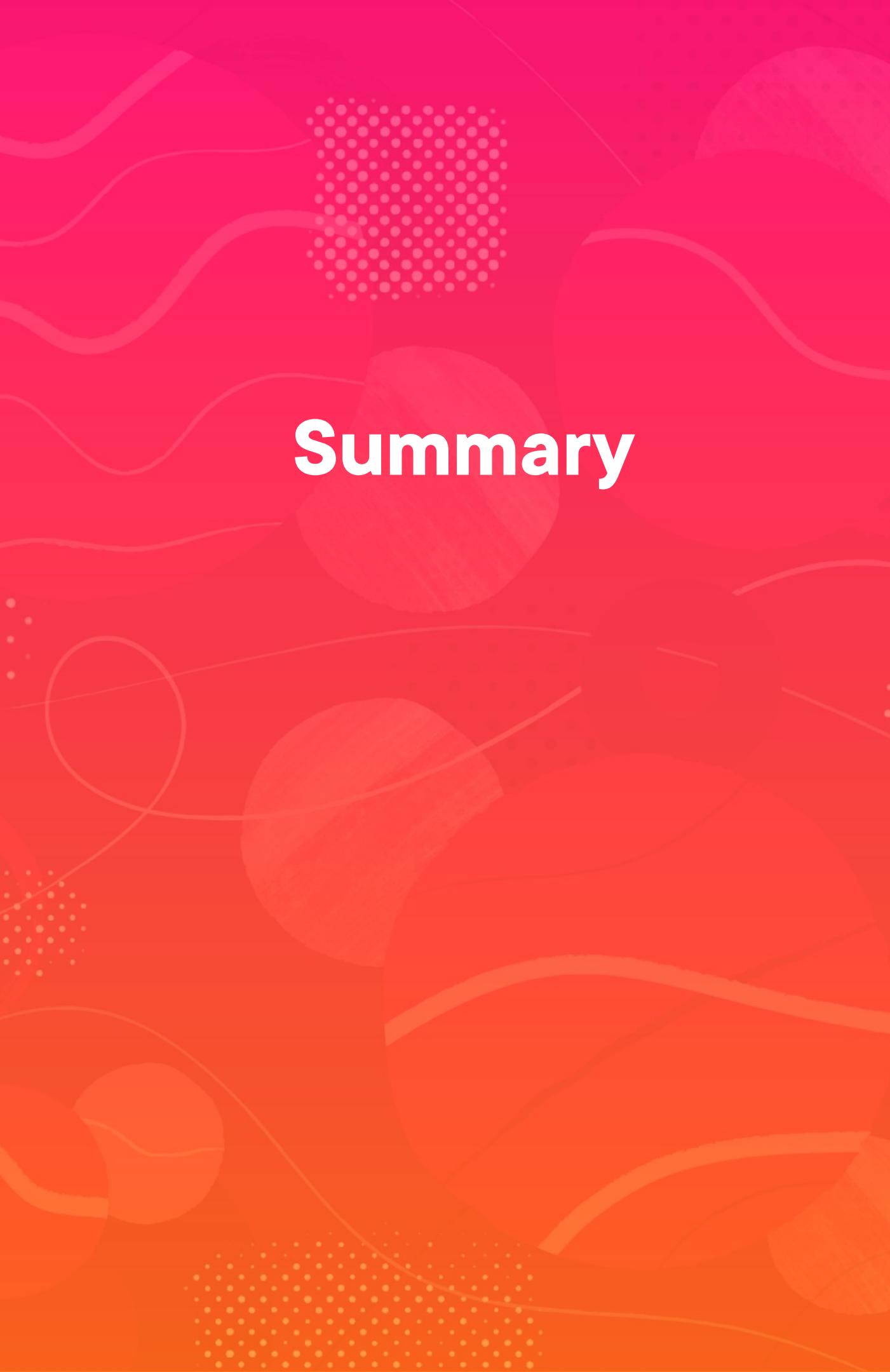
# Options Store

```
defineStore('cart', {  
  state: () => ({ cart: [] }),  
  
  getters: {  
    cartTotal: (state) => { ... };  
  },  
  
  actions: {  
    fetchCartData() {  
      ...  
    }  
  },  
});
```

# Setup Store

```
defineStore('cart', () => {  
  const cart = ref([]);  
  
  const cartTotal = computed(() =>  
    {  
      return cart.reduce(...);  
    }  
  );  
  
  fetchCartData() {  
    ...  
  };  
  
  return {cart, cartTotal, fetchCartData };  
});
```





# Summary

- What is state management?
- Creating a Pinia store
- Storing and accessing data in a Pinia store
- Managing reactivity with store state
- Using calculated data (getters)
- Using actions to fetch data from an API
- Setup stores vs Options stores
- Accessing stores from Composition/Options API components



**Up Next:**

# **Creating Custom Vue Directives**

---

