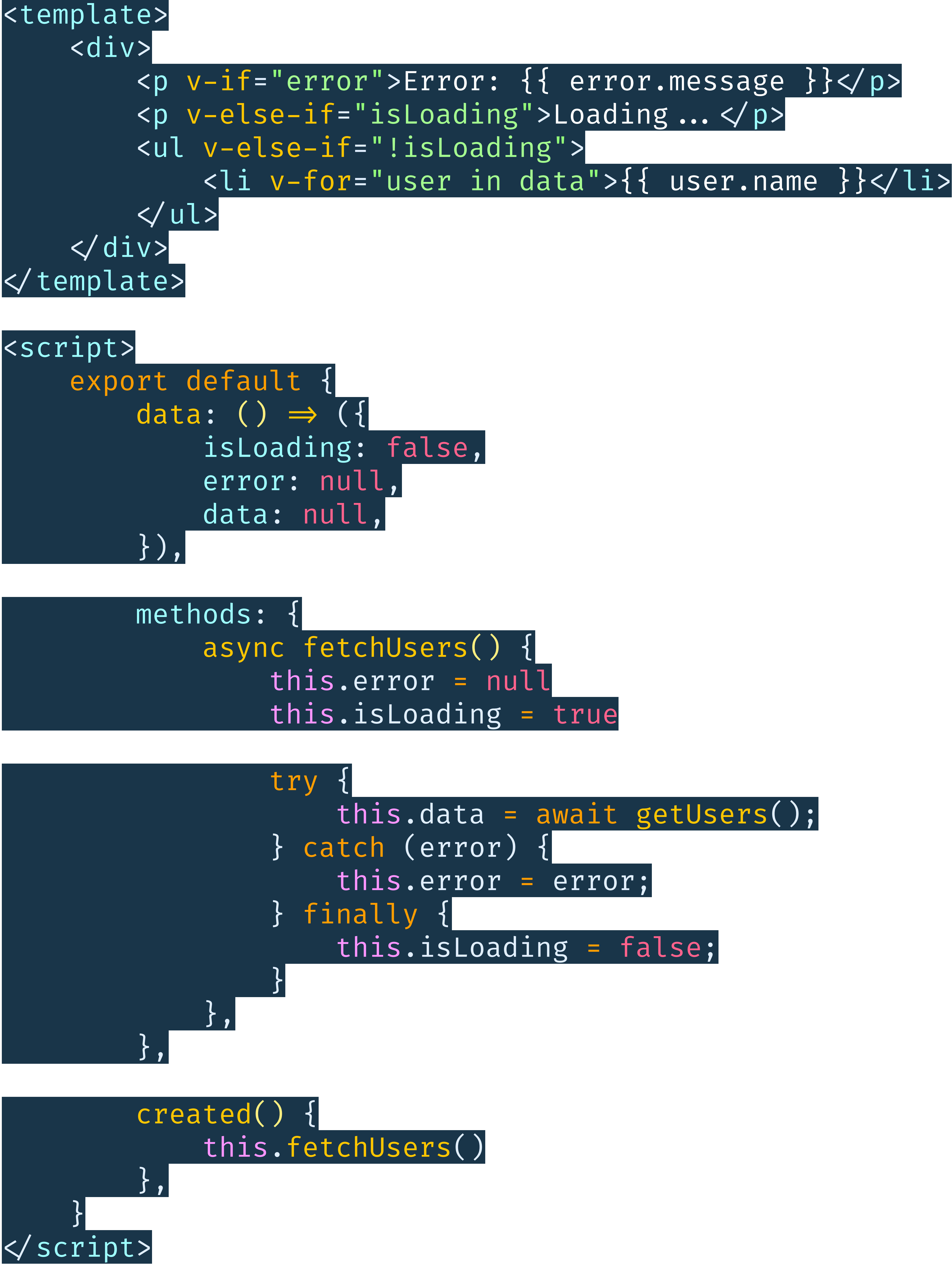
vue-promised



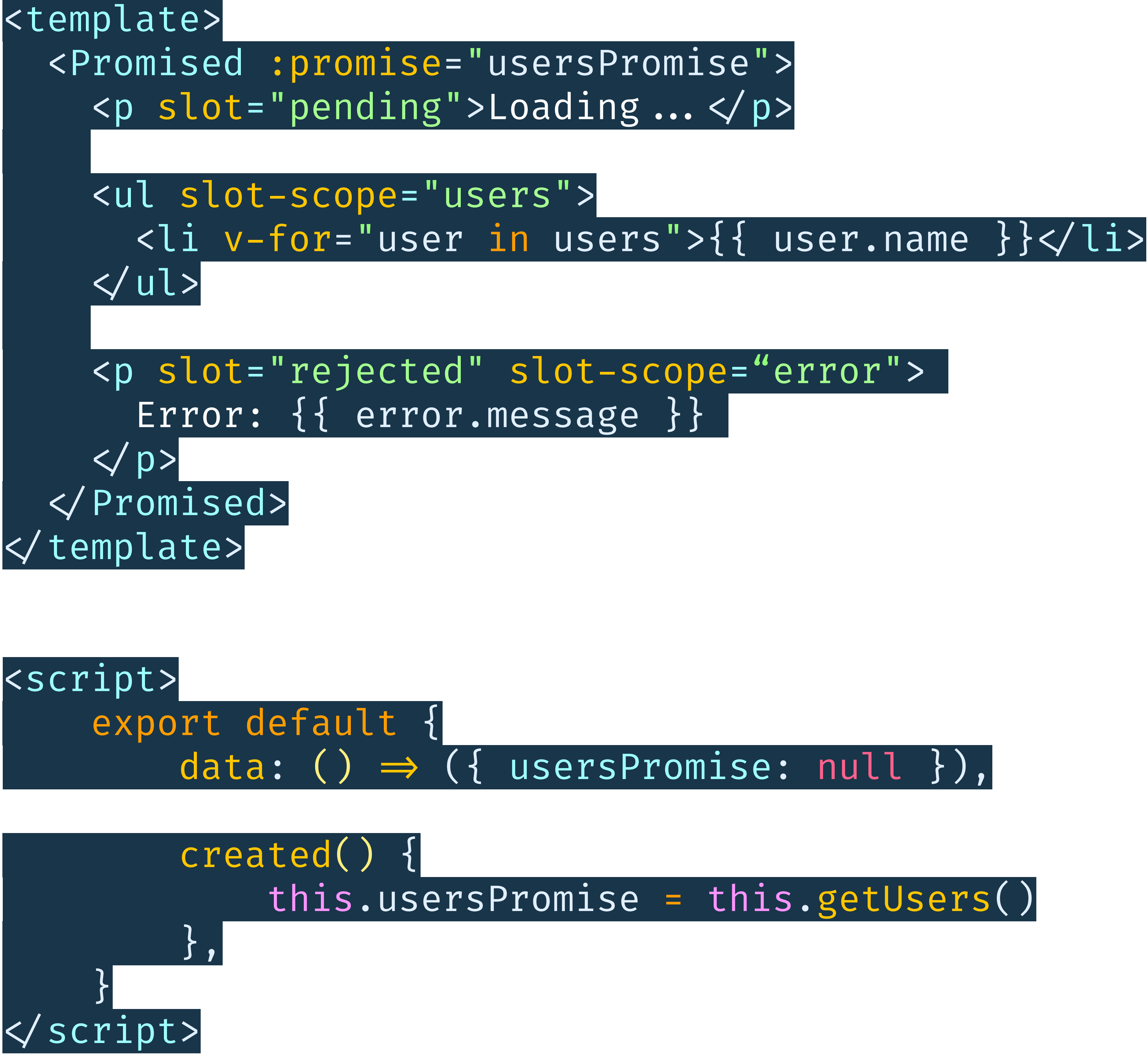
```
<template>
    <div>
        <p v-if="error">Error: {{ error.message }}</p>
        <p v-else-if="isLoading">Loading...</p>
        <ul v-else-if="!isLoading">
            <li v-for="user in data">{{ user.name }}</li>
        </ul>
    </div>
</template>

<script>
    export default {
        data: () => ({
            isLoading: false,
            error: null,
            data: null,
        }),

        methods: {
            async fetchUsers() {
                this.error = null
                this.isLoading = true

                try {
                    this.data = await getUsers();
                } catch (error) {
                    this.error = error;
                } finally {
                    this.isLoading = false;
                }
            },
        },

        created() {
            this.fetchUsers()
        },
    }
</script>
```



```
<template>
  <Promised :promise="usersPromise">
    <p slot="pending">Loading...</p>

    <ul slot-scope="users">
      <li v-for="user in users">{{ user.name }}</li>
    </ul>

    <p slot="rejected" slot-scope=“error">
      Error: {{ error.message }}
    </p>
  </Promised>
</template>


<script>
    export default {
        data: () => ({ usersPromise: null }),

        created() {
            this.usersPromise = this.getUsers()
        },
    }
</script>
```

```
<template>
    <div>
        <p v-if="error">Error: {{ error.message }}</p>
        <p v-else-if="isLoading">Loading...</p>
        <ul v-else-if="!isLoading">
            <li v-for="user in data">{{ user.name }}</li>
        </ul>
    </div>
</template>

<script>
    export default {
        data: () => ({
            isLoading: false,
            error: null,
            data: null,
        }),

        methods: {
            async fetchUsers() {
                this.error = null
                this.isLoading = true

                try {
                    this.data = await getUsers();
                } catch (error) {
                    this.error = error;
                } finally {
                    this.isLoading = false;
                }
            },
        },

        created() {
            this.fetchUsers()
        },
    }
</script>
```

```
<template>
  <Promised :promise="usersPromise">
    <p slot="pending">Loading...</p>

    <ul slot-scope="users">
      <li v-for="user in users">{{ user.name }}</li>
    </ul>

    <p slot="rejected" slot-scope=“error">
      Error: {{ error.message }}
    </p>
  </Promised>
</template>


<script>
    export default {
        data: () => ({ usersPromise: null }),

        created() {
            this.usersPromise = this.getUsers()
        },
    }
</script>
```

vue-promised

portal-vue