Class 5: Props & Slots

- Review: Last Class
- Props
- Slots
- Component Styling

Review: Last Class

Review: Event Handling

The `v-on` directive listens for DOM events and runs JavaScript code when they're triggered.

```
<script setup>
import { ref } from 'vue'
const count = ref(0)

function incrementCount() {
  count.value++
}
</script>
```

```
<template>
  <button v-on:click="incrementCount">
    {{ count }}
  </button>
</template>
```

Review: Event Handlers

In response to a DOM event, Vue.js can run a snippet of JavaScript code or call a function handler.

Handler Function

```
v-on:click="incrementCount"
```

Inline Code

```
v-on:click="count++"
```

Not a Handler:

```
v-on:click"incrementCount()"
```

Review: Conditional Rendering

The `v-if`, `v-else-if`, `v-else`, and `v-show` directives conditionally render elements based on **JavaScript expressions**.

```
<script setup>
import { ref } from 'vue'
const isVisible = ref(true)
</script>
<template>
  <div v-if="isVisible">
    This element is visible.
  </div>
</template>
```

Homework 2: Parts III, IV, && V

Discussion: Homework 2 - Parts I && II

- Part III: Props
- Part IV: Slots
- Part V: Styled Prototype

Props

Props

Custom attributes that allow JavaScript data to be passed from a parent component to a child component.

Defining Props

`ChildComponent.vue`:

```
<script setup>
defineProps(['propName1', 'propName2', ...])
</script>
```

Passing Props

```
<script setup>
import ChildComponent
  from './components/ChildComponent.vue'
</script>
<template>
  <ChildComponent
    propName1="sample string 1"
    propName2="sample string 2" />
</template>
```

Example: Props

`Book.vue`:

```
<script setup>
defineProps(['title', 'author'])
</script>
<template>
  <hgroup>
    <h2>{{ title }}</h2>
    <p>by {{ author }}</p>
  </hgroup>
</template>
```

`App.vue`:

```
<script setup>
import Book from './components/Book.vue'
</script>
<template>
  <Book
    title="The Great Gatsby"
    author="F. Scott Fitzgerald" />
  <Book
    title="1984"
    author="George Orwell" />
</template>
```

Gotcha: Props are Read-Only

Data flows from parent to child only.

Child components cannot modify prop values.

Gotcha: Default Value is a String

```
<ChildComponent propName="42" />
```

`propName` will be a string with the value `"42"`, not a number with the value `42`.

To pass other data types, we'll need to bind the prop using `v-bind` or the `:` shorthand. (Future class)

Slots

Slots

Placeholders inside a child component that allow HTML content to be passed from a parent component to a child component.

Defining Slots

`ChildComponent.vue`:

```
<template>
  <slot></slot>
</template>
```

Passing Slot Content

```
<script setup>
import ChildComponent
  from './components/ChildComponent.vue'
</script>
<template>
  <ChildComponent>
    <p>This is slot content.</p>
  </ChildComponent>
</template>
```

Example: Slots

`AlertBox.vue`:

```
<template>
  <div class="alert-box">
    <slot></slot>
  </div>
</template>
```

`App.vue`:

```
<script setup>
import AlertBox
  from './components/AlertBox.vue'
</script>
<template>
  <AlertBox>
    <h2>Warning!</h2>
    <p>This action cannot be undone.</p>
  </AlertBox>
</template>
```

Named Slots

`ChildComponent.vue`:

```
<template>
  <slot name="header"></slot>
  <slot></slot>
  <slot name="footer"></slot>
</template>
```

`App.vue`:

```
<ChildComponent>
  <template v-slot:header>
    <h2>This is the header slot content.</h2>
  </template>
  <p>This is the default slot content.</p>
  <template v-slot:footer>
    <p>This is the footer slot content.</p>
  </template>
</ChildComponent>
```

Shorthand: `v-slot`

`#` can be used as a shorthand for `v-slot:`

```
<ChildComponent>
  <template #header>
    <h2>This is the header slot content.</h2>
  </template>
  <p>This is the default slot content.</p>
  <template #footer>
    <p>This is the footer slot content.</p>
  </template>
</ChildComponent>
```

```
<ChildComponent>
  <template v-slot:header>
    <h2>This is the header slot content.</h2>
  </template>
  <p>This is the default slot content.</p>
  <template v-slot:footer>
    <p>This is the footer slot content.</p>
  </template>
</ChildComponent>
```

Activity: Props & Slots Practice

Working with your peers (2-4), correct the mistakes in the provided `Menu.vue` and `App.vue` components to implement a working hamburger menu using props and slots.

Vue.js Styling

Styling in Vue.js

Standard HTML elements can be styled.

```
<template>
  <h1 class="title">
    Hello, World!
  </h1>
</template>

<style scoped>
.title {
  color: blue;
}
</style>
```

Scoped Styles

By default, styles defined in a Vue component are scoped to that component only.

```
<style scoped>
.title {
  color: blue;
}
</style>
```

This prevents styles from leaking out to other components or the global scope.

Gotcha: Unscoped Styles

If the `scoped` attribute is omitted, styles will apply globally to **all** components.

```
<style>
.title {
  color: blue;
}
</style>
```

Any element with the `title` class in **any** component will be blue.

Global Styles

Global styles can be defined in the `src/assets` folder and imported in `main.js`:

```
import './assets/main.css'
```

- `/.assets/base.css` - Base styles for the entire app
- `/.assets/main.css` - Main styles for the entire app

Gotcha: Vue.js Component’s Cannot Use the Element Selector

```
<template>
  <h1>Hello</h1>
  <ChildComponent />
</template>
```

Only proper HTML elements can use the element selector: `h1 { ... }`

Vue.js components are not proper HTML elements and cannot use the element selector: `ChildComponent { ... }`

Studio: Homework 2

Working with your peers (2-4), complete Parts III, IV, and V of Homework 2.

Summary

- Props allow data to be passed from parent to child components.
- Slots allow HTML content to be passed from parent to child components.
- Styles can be scoped to individual components or applied globally.