Сучасні каркаси web-додатків

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Як зібрати перевикористовуваний компонент, який керує розміткою, розміщенням і фокусом

Лекція 3 · Лісняк Андрій Олександрович

Вікно, яке не залежить від свого вмісту

Розділити

Визначити, що передається пропсом, а що слотом, і чому це різні механізми.

Передати

Використати слот за замовчуванням, іменований слот і слот із параметрами.

Дістатися DOM

Оголосити шаблонне посилання й дочекатися оновлення дерева документа.

Розмістити

Вивести панель у body через Teleport, зберігши логічну ієрархію.

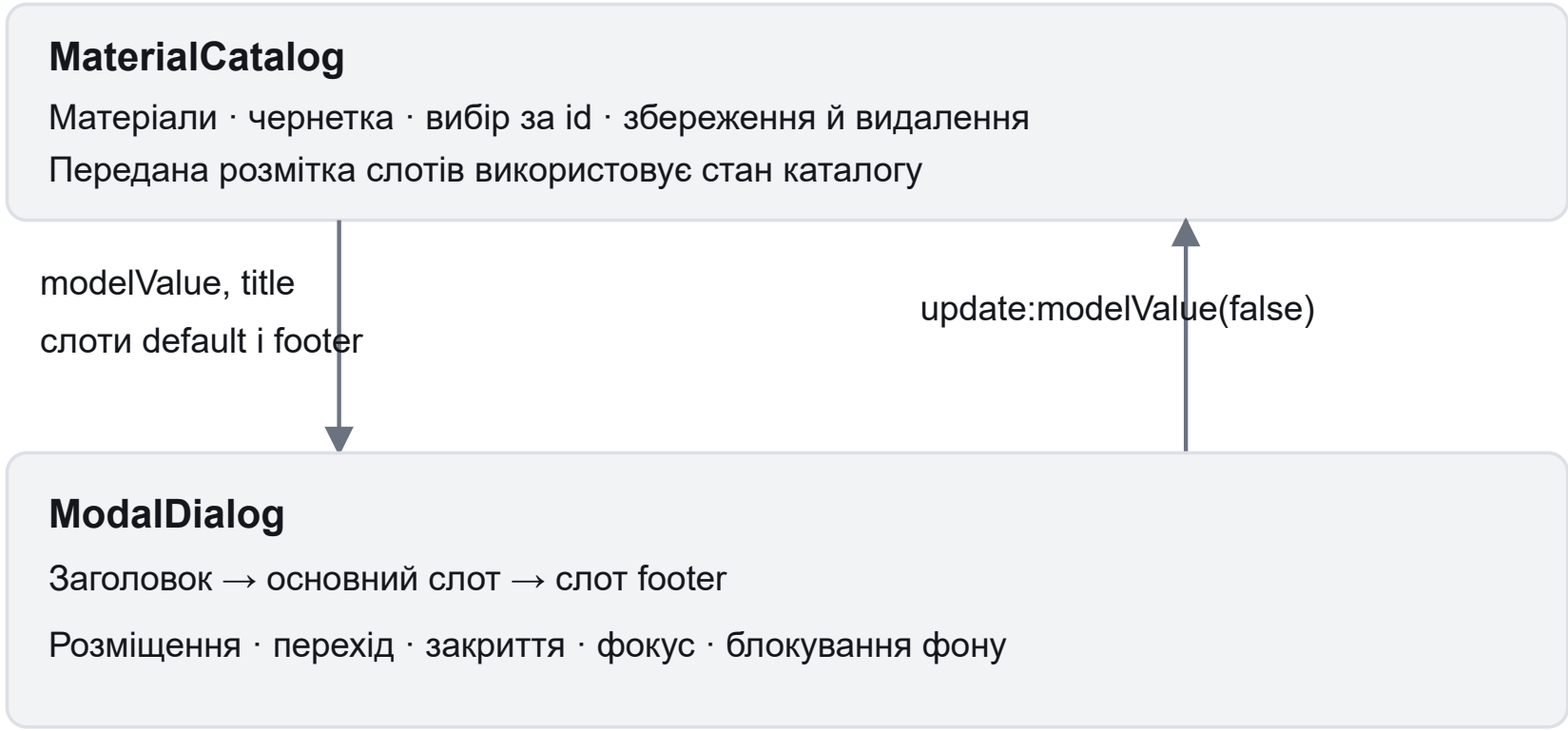
Анімувати

Узгодити появу та зникнення з Transition, не втративши керування станом.

Утримати фокус

Перенести фокус у вікно, заблокувати фон і повернути фокус після закриття.

Каталог володіє даними, діалог — поведінкою вікна



Вниз ідуть значення й розмітка, вгору — єдина подія про намір закритися.

Значення передають пропсом, розмітку — слотом

Що передаємо	Яким механізмом
Заголовок вікна	Пропс title: рядок, який компонент виводить сам
Ознака відкритості	Пропс modelValue та подія update:modelValue
Форма з полями й підписами	Слот за замовчуванням: розмітка належить батькові
Кнопки поточного сценарію	Іменований слот footer
Рядок списку зі станом дочірнього компонента	Слот із параметрами

Слот — це зарезервоване місце для чужої розмітки

Компонент відводить місце, а заповнює його той, хто компонент підключає.

Слот

Місце у шаблоні компонента, куди підставляється розмітка від батьківського компонента.

Точка виведення

Тег slot у шаблоні дочірнього компонента; на його місце стає переданий вміст.

Запасний вміст

Розмітка між тегами slot і /slot. Показується, доки нічого не передано.

> Джерело: Vue.js · Слоти — https://ua.vuejs.org/guide/components/slots.html

Панель володіє оболонкою, а не вмістом

src/components/InfoPanel.vue · рядки 9–23

```
<template>
  <section class="panel">
    <h2 class="panel__title">{{ title }}</h2>

    <div class="panel__body">
      <!-- Слот за замовчуванням приймає весь неіменований вміст -->
      <slot>Вміст не передано.</slot>
    </div>

    <footer class="panel__footer">
      <!-- Іменований слот виводить розмітку, передану через <template #footer> -->
```

`npm install && npm run dev`

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Обробник належить батькові, кнопка — панелі

src/App.vue · рядки 22–34

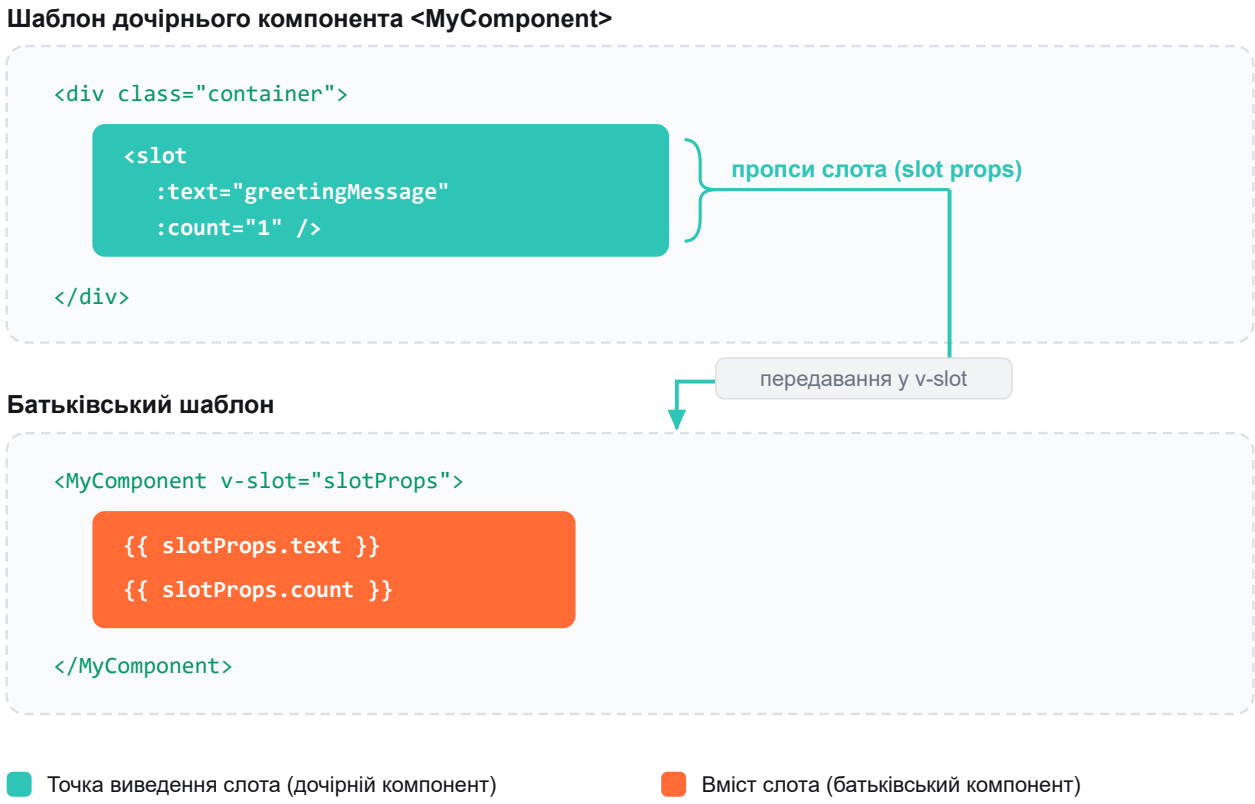
```
    <InfoPanel title="Нотатка до заняття">
      <label class="field">
        Текст нотатки
        <input v-model="note" type="text">
      </label>
      <p v-if="saved" class="saved">Збережено: {{ saved }}</p>

      <template #footer>
        <!-- Обробник належить цьому компоненту, хоча кнопка показана в панелі -->
        <button type="button" @click="note = ''">Очистити</button>
        <button type="button" @click="save">Зберегти</button>
      </template>
    </InfoPanel>
```

```
npm install && npm run dev
```

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Слот із параметрами передає дані в шаблон батька



> Дочірній компонент віддає локальні значення, батьківський вирішує, як їх показати.

Список володіє даними, рядок малює батько

Дочірній · src/components/TaskList.vue · рядки 19–27

```
    <li v-for="task in tasks" :key="task.id" class="list__item">
      <!-- Пропси слота: батько отримає task і toggle -->
      <slot
        name="row"
        :task="task"
        :toggle="() => toggle(task)"
      >
        {{ task.title }}
      </slot>
```

Батьківський · src/App.vue · рядки 38–47

```
      <TaskList>
        <!-- Слот із параметрами: task і toggle з TaskList -->
        <template #row="{ task, toggle }">
          <label class="row">
            <input
              type="checkbox"
              :checked="task.done"
              @change="toggle"
            >
            <span :class="{ 'row--done': task.done }">
```

```
npm install && npm run dev
```

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Два різні ref розв'язують дві різні задачі

Механізм	Що зберігає
ref(false)	Реактивне значення. Зміна .value спричиняє оновлення шаблону
ref(null) з атрибутом ref	Шаблонне посилання на змонтований елемент DOM або екземпляр компонента
Функція у :ref	Викликається під час монтування й оновлення; одержує елемент або null при демонтажі
Посилання як значення пропса	Одержаний елемент передають далі звичайним пропсом, наприклад initial-focus

> Джерело: Vue.js · Посилання на шаблон — https://ua.vuejs.org/guide/essentials/template-refs.html

Змінна й атрибут ref — це одне посилання

script · src/components/BaseModal.vue · рядки 12–16

```
const panel = ref(null)

// Елемент, який відкрив вікно: на нього повертається фокус після закритт

let opener = null

let background = null

let previousOverflow = ''
```

шаблон · src/components/BaseModal.vue · рядки 53–61

```
        <section

          ref="panel"

          class="panel"

          role="dialog"

          aria-modal="true"

          aria-labelledby="base-modal-title"

          tabindex="-1"

          @keydown.esc="close"

        >
```

```
npm install && npm run dev
```

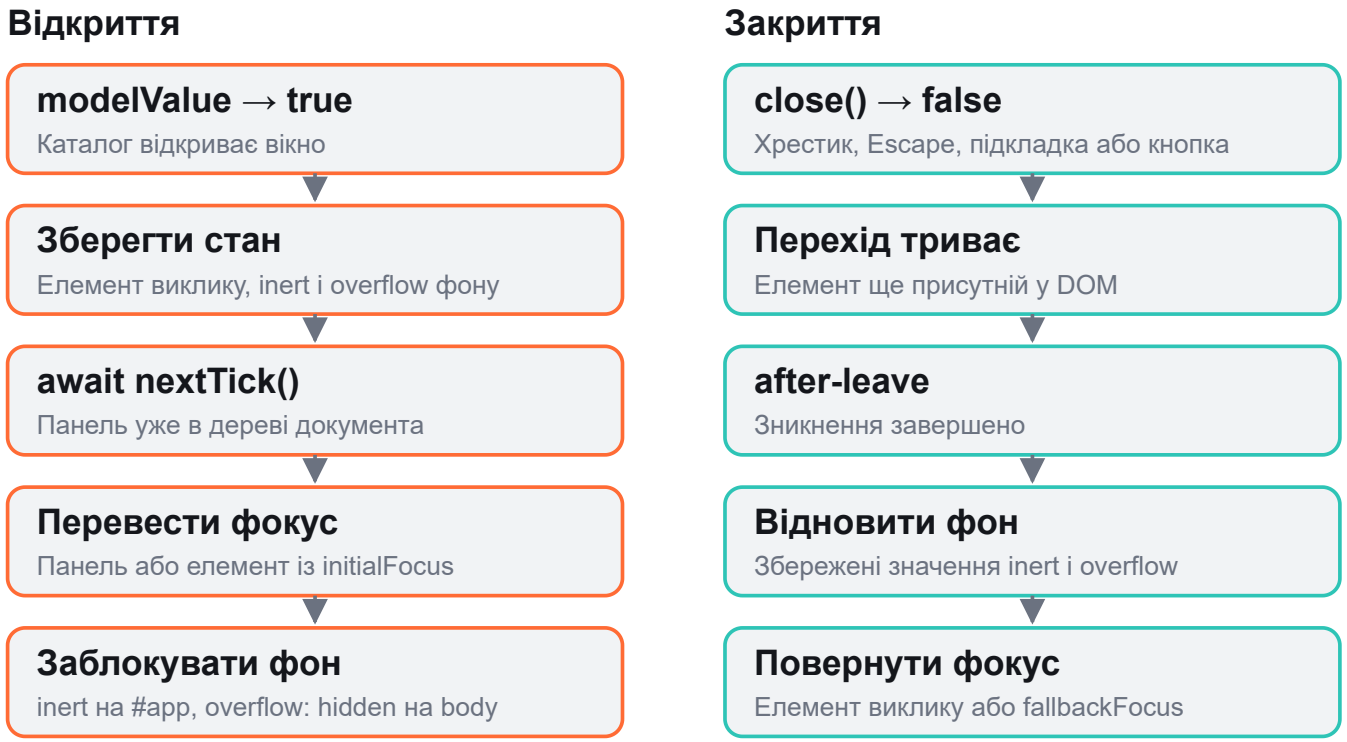
Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

DOM оновлюється не в тому самому рядку коду

1. **Змінили стан** Присвоєння modelValue не перебудовує дерево документа негайно.
2. **Vue накопичує** Зміни застосовуються однією партією після поточної синхронної роботи.
3. **await nextTick()** Виклик завершується, коли Vue вже оновило дерево документа.
4. **Браузер малює** Перерахунок геометрії та малювання — окремий етап, nextTick його не чекає.

Джерело: Vue.js · nextTick — https://ua.vuejs.org/api/general.html#nexttick

Порядок відкриття й закриття фіксований



Ту саму процедуру очищення виконує onBeforeUnmount: події after-leave при демонтажі не буде.

> Фон блокують після перенесення фокуса, а відновлюють після завершення зникнення.

Панель усередині картки підкоряється предкам

Три властивості предка ламають розміщення, і жодну з них не видно з коду самого вікна.

transform

Предок із transform, perspective або filter стає системою координат для position fixed.

overflow

Предок із прихованим переповненням обрізає те, що вийшло за його межі.

Контекст накладання

Велике z-index не піднімає панель над елементами поза її контекстом.

> Джерело: Vue.js · Teleport — https://ua.vuejs.org/guide/built-ins/teleport.html

Картку навмисно зроблено ворожою до панелі

src/App.vue · рядки 62–70

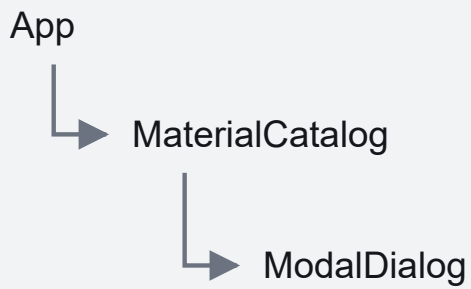
```
/* Саме ці дві властивості й створюють проблему, яку розв'язує Teleport */
.card {
  transform: translateZ(0);
  overflow: hidden;
  padding: 20px;
  border: 1px solid #c8c3b4;
  border-radius: 12px;
  background: #fff;
}
```

`npm install && npm run dev`

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Ієрархія зберігається, змінюється місце в DOM

Дерево компонентів



Пропси й події збережено

Слоти належать каталогу

DOM після Teleport



Діалог розміщено поза #app

Пропси, події та слоти працюють так само, хоча вузол лежить поза контейнером застосунку.

Умова живе всередині компонента, а не зовні

src/components/BaseModal.vue · рядки 48–61

```
<template>
  <!-- Панель виводиться в body, тому обмеження предків на неї не діють -->
  <Teleport to="body">
    <Transition name="fade" @after-leave="release">
      <div v-if="modelValue" class="overlay" data-testid="overlay" @click.self="close">
        <section
          ref="panel"
          class="panel"
          role="dialog"
          aria-modal="true"
          aria-labelledby="base-modal-title"
```

`npm install && npm run dev`

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Transition дає класи, а не анімацію

`enter-from`	Початковий стан появи; знімається в наступному кадрі.
`enter-active`	Діє протягом усієї появи; саме тут задають тривалість і функцію згладжування.
`enter-to`	Кінцевий стан появи.
`leave-*`	Ті самі три ролі — початок, дія, кінець — для зникнення.
`after-enter, after-leave`	Події про завершення переходу; місце для фокуса та очищення.

> Джерело: Vue.js · Transition — https://ua.vuejs.org/guide/built-ins/transition.html

Фон відновлюють після зникнення і при демонтажі

src/components/BaseModal.vue · рядки 34–45

```
// Фон розблоковується після завершення переходу, а не разом зі зміною стану.
function release() {
  if (background) background.inert = false
  document.body.style.overflow = previousOverflow
  if (opener instanceof HTMLElement && opener.isConnected) {
    opener.focus({ preventScroll: true })
  }
}


// Ту саму процедуру виконуємо під час демонтажу: інакше сторінка
// лишиться заблокованою, якщо компонент прибрали у відкритому стані.
```

`npm install && npm run dev`

Модальне вікно: слоти, шаблонні посилання, Teleport і Transition

Атрибути описують семантику, а не поведінку

role="dialog" і aria-modal="true"

Повідомляють допоміжним технологіям, що це діалог, який перекриває решту сторінки.

aria-labelledby

Пов'язує панель із її заголовком; заголовок одержує унікальний id.

Атрибут inert

Вилучає фон із взаємодії та з дерева доступності. Ставиться на контейнер застосунку, не на body.

Блокування прокручування

Попереднє значення overflow зберігають, щоб повернути саме його, а не порожній рядок.

Джерело: WAI-ARIA APG · Dialog Modal — https://www.w3.org/WAI/ARIA/apg/patterns/dialog-modal/

Фокус входить, лишається й повертається назад

1. **Запам'ятати** Зберегти активний елемент до відкриття — це кнопка виклику.
2. **Перевести** Після оновлення DOM спрямувати фокус у поле форми або на кнопку скасування.
3. **Утримати** Обробити Tab і Shift+Tab по колу серед видимих доступних елементів панелі.
4. **Повернути** Після закриття — на кнопку виклику, а якщо вона зникла, на запасну ціль.

Панель має вміщатися в екран, а не в задум

`max-h-[calc(100dvh-2rem)]`	Обмеження за поточною висотою області перегляду з полем на відступи.
`min-h-0` та `overflow-y-auto`	Прокручується лише основний вміст; без обнуленої мінімальної висоти flex його не стисне.
`shrink-0`	Заголовок і нижня область не стискаються, тому кнопки лишаються видимими.
`flex-col` та `sm:flex-row`	Кнопки стають у рядок від порога 640 px у типовій конфігурації Tailwind.
`min-h-11` та `min-w-11`	Область натискання щонайменше 44 px; поля тієї самої висоти.

Що далі

Дорожня карта

1. **Слоти** Пропс передає значення, слот — розмітку; пропси слота повертають дані в шаблон батька.
2. **Посилання** Шаблонний ref дає елемент, nextTick — момент, коли він уже в дереві документа.
3. **Розміщення й час** Teleport виводить панель у body, Transition узгоджує появу та зникнення.
4. **Фокус і фон** Фон блокують після перенесення фокуса, відновлюють після завершення зникнення.

Завдання

- Лабораторна робота 3: один ModalDialog для редагування та підтвердження видалення
- Прочитати розділи гіда про слоти, Teleport і Transition українською
- Підготувати питання на допуск за темою modal-dialog

Дякую!

Ваші запитання