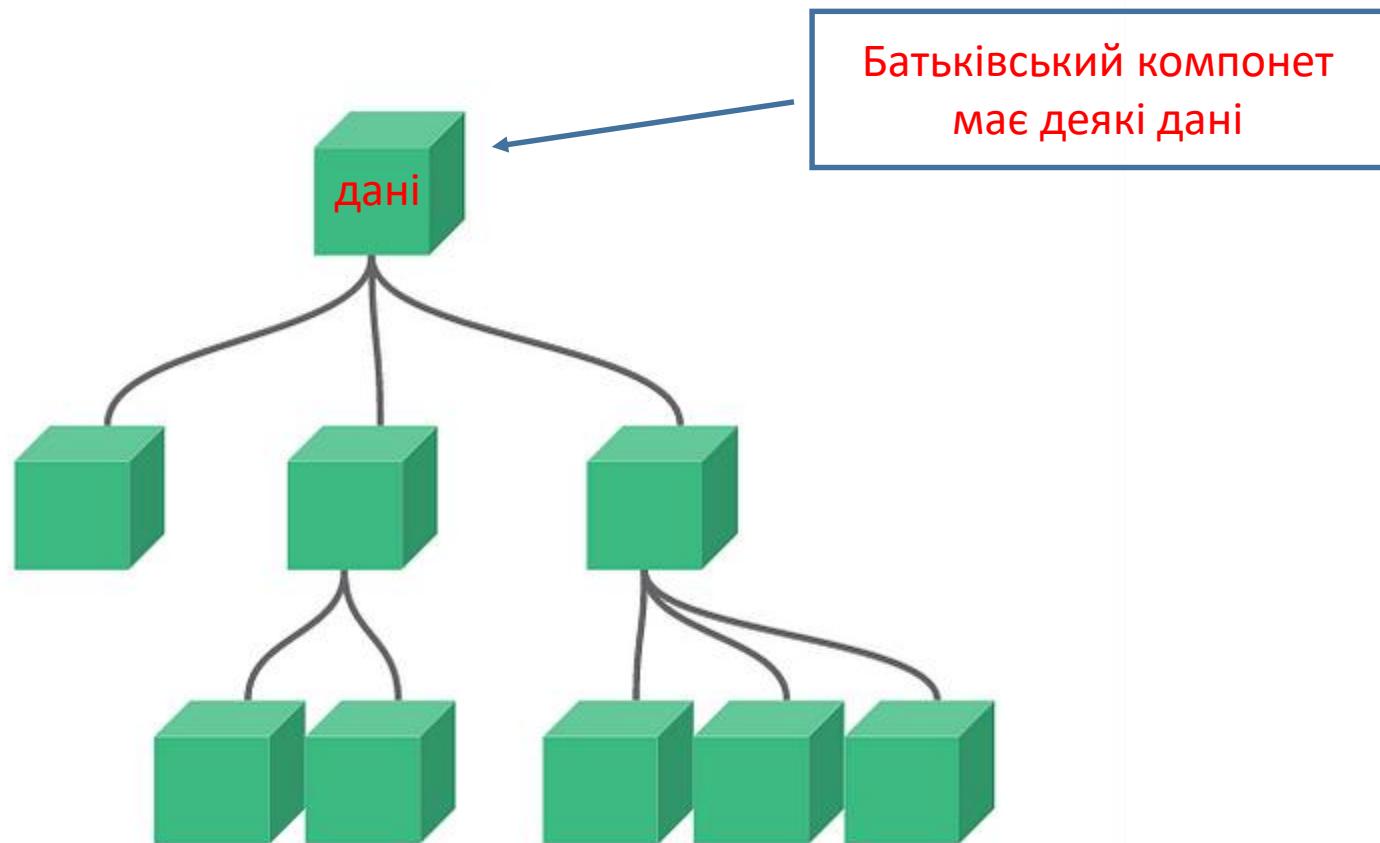


Поширення даних для дочірніх компонентів

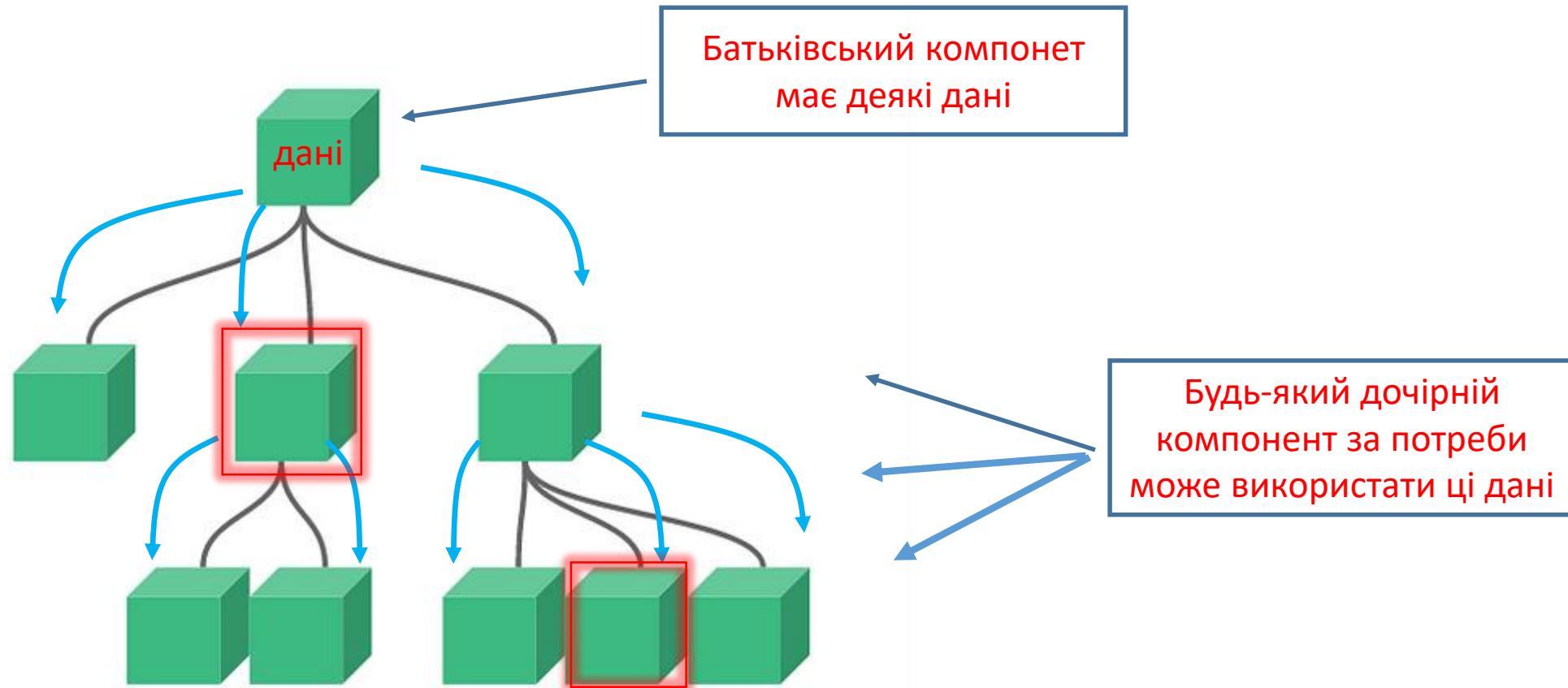
provide / inject

<https://ua.vuejs.org/guide/components/provide-inject.html>

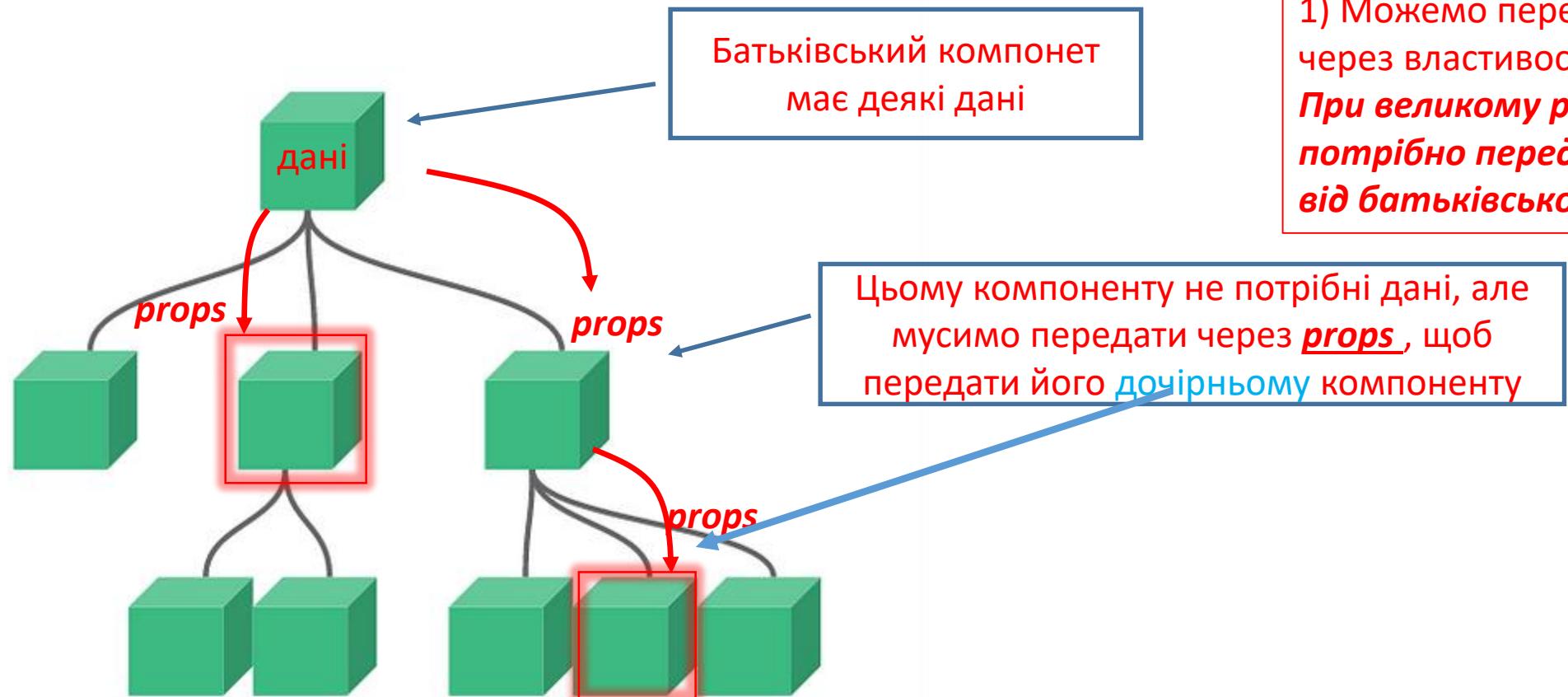
Що робити, якщо маємо вкладені компоненти і потрібно передати дані від батьківського до дочірніх?



Що робити, якщо маємо вкладені компоненти і потрібно передати дані від батьківського до дочірніх?

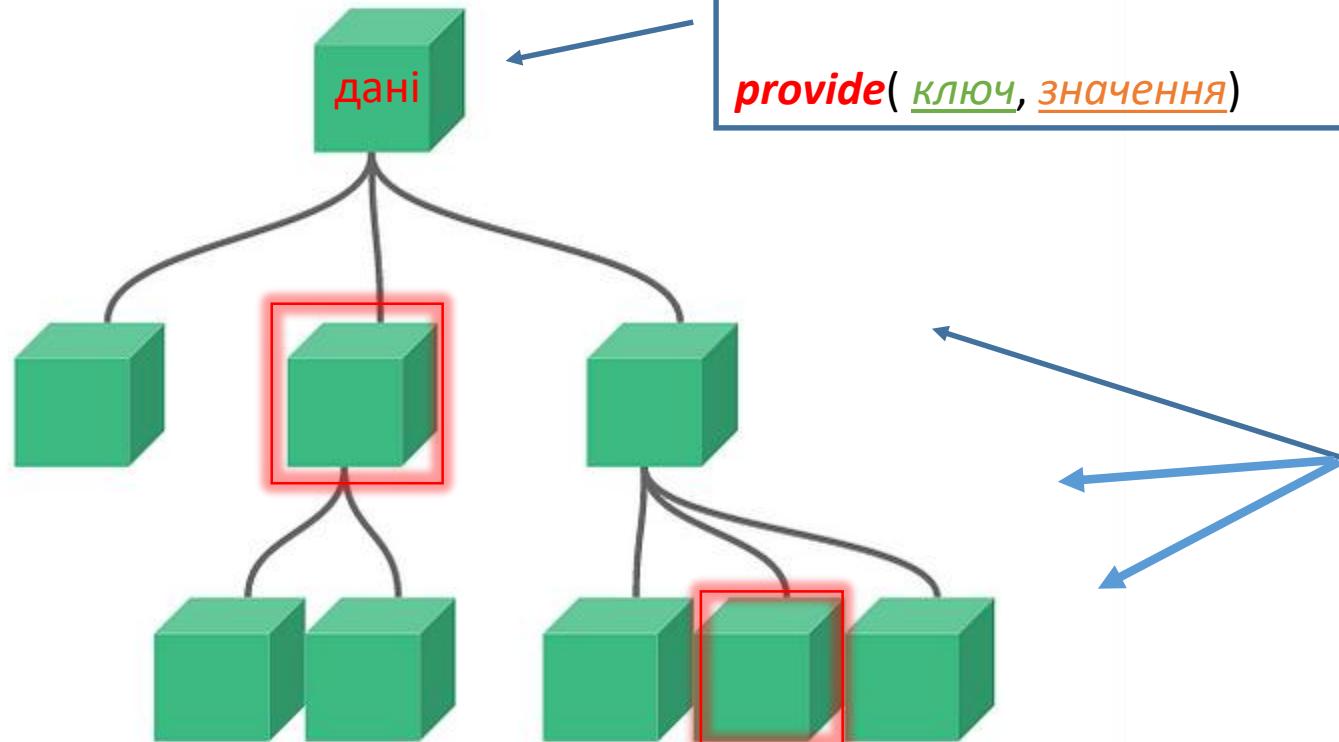


Що робити, якщо маємо вкладені компоненти і потрібно передати дані від батьківського до дочірніх?



1) Можемо передавати
через властивості *props*
При великому рівні вкладеності
потрібно передавати багато разів
від батьківського до дочірнього

Що робити, якщо маємо вкладені компоненти і потрібно передати дані від батьківського до дочірніх?



Батьківський надає дані
import { **provide** } from 'vue'
provide(ключ, значення)

2) **provide | inject**

- батьківський “надає дані” (**provide**)
- будь-який дочірній за потреби «додає собі дані» (**inject**)

Будь-який дочірній додає собі ці дані
import { **inject** } from 'vue'
const внутрішня_назва = **inject**(ключ)

▼ ParentComp.vue ProvidelInject ●

```
Comp.vue > {} "ParentComp.vue" > template > div
1  <template>
2  ...<div>
3  ....<h1>Parent Component</h1>
4  ....<p>isActive:<br/>
5  .....<input v-model="isActive">
6  .....type="checkbox"/>
7  ....</p>
8  ...<MediumComp />
9  </div>
10 </template>
11 <script setup>
12 import { ref, provide } from 'vue'
13 import MediumComp from './
MediumComp.vue'
14
15 const isActive = ref(true)
16 provide('isActive', isActive)
17 </script>
```

provide('isActive', isActive) ← - - - PrentComp

▼ MediumComp.vue ProvidelInject ●

```
using-test > src > components > ProvidelInject > ▼ MediumComp.vue
1  <template>
2  ...<div>
3  ....<h2>Medium Component</h2>
4  ....<!-- Не використовуємо isActive у цьому компоненті -->
5  ...<ChildComp />
6  ...</div>
7  </template>
8
9  <script setup>
10 import ChildComp from './ChildComp.vue'
11 </script>
```

MediumComp

ChildComp

inject('isActive')

▼ ChildComp.vue ProvidelInject X

```
using-test > src > components > ProvidelInject > ▼ ChildComp.vue > {}
1  <template>
2  ...<div>
3  ....<h3>Child Component</h3>
4  ....<p>isActive from Parent: {{ isActive }}</p>
5  ...</div>
6  </template>
7
8  <script setup>
9  import { inject } from 'vue'
10 const isActive = inject('isActive')
11 </script>
```

▼ ParentComp.vue ProvidelInject

```
Comp.vue > {} "ParentComp.vue" > template > div
1  <template>
2  ...<div>
3  ....<h1>Parent Component</h1>
4  ....<p>isActive:<br/>
5  .....<input v-model="isActive">
6  .....type="checkbox"/>
7  ....</p>
8  ...<MediumComp />
9  </div>
10 </template>
11 <script setup>
12 import { ref, provide } from 'vue'
13 import MediumComp from './
MediumComp.vue'
14
15 const isActive = ref(true)
16 provide('isActive', isActive)
17 </script>
```

provide('isActive', isActive) ← - - - PrentComp

▼ MediumComp.vue ProvidelInject

```
using-test > src > components > ProvidelInject > ▼ MediumComp.vue
1  <template>
2  ...<div>
3  ....<h2>Medium Component</h2>
4  ....<!-- Не використовуємо isActive у цьому компоненті -->
5  ...<ChildComp />
6  ...</div>
7  </template>
8
9  <script setup>
10 import ChildComp from './ChildComp.vue'
11 </script>
```

provide('isActive', isActive)

MediumComp

ChildComp

inject('isActive')

▼ ChildComp.vue ProvidelInject

```
using-test > src > components > ProvidelInject > ▼ ChildComp.vue > {}
1  <template>
2  ...<div>
3  ....<h3>Child Component</h3>
4  ....<p>isActive from Parent: {{ isActive }}</p>
5  ...</div>
6  </template>
7
8  <script setup>
9  import { inject } from 'vue'
10 const isActive = inject('isActive')
11 </script>
```

Parent Component

isActive:

Medium Component

Child Component

isActive from Parent: true

Parent Component

isActive:

Medium Component

Child Component

isActive from Parent: false