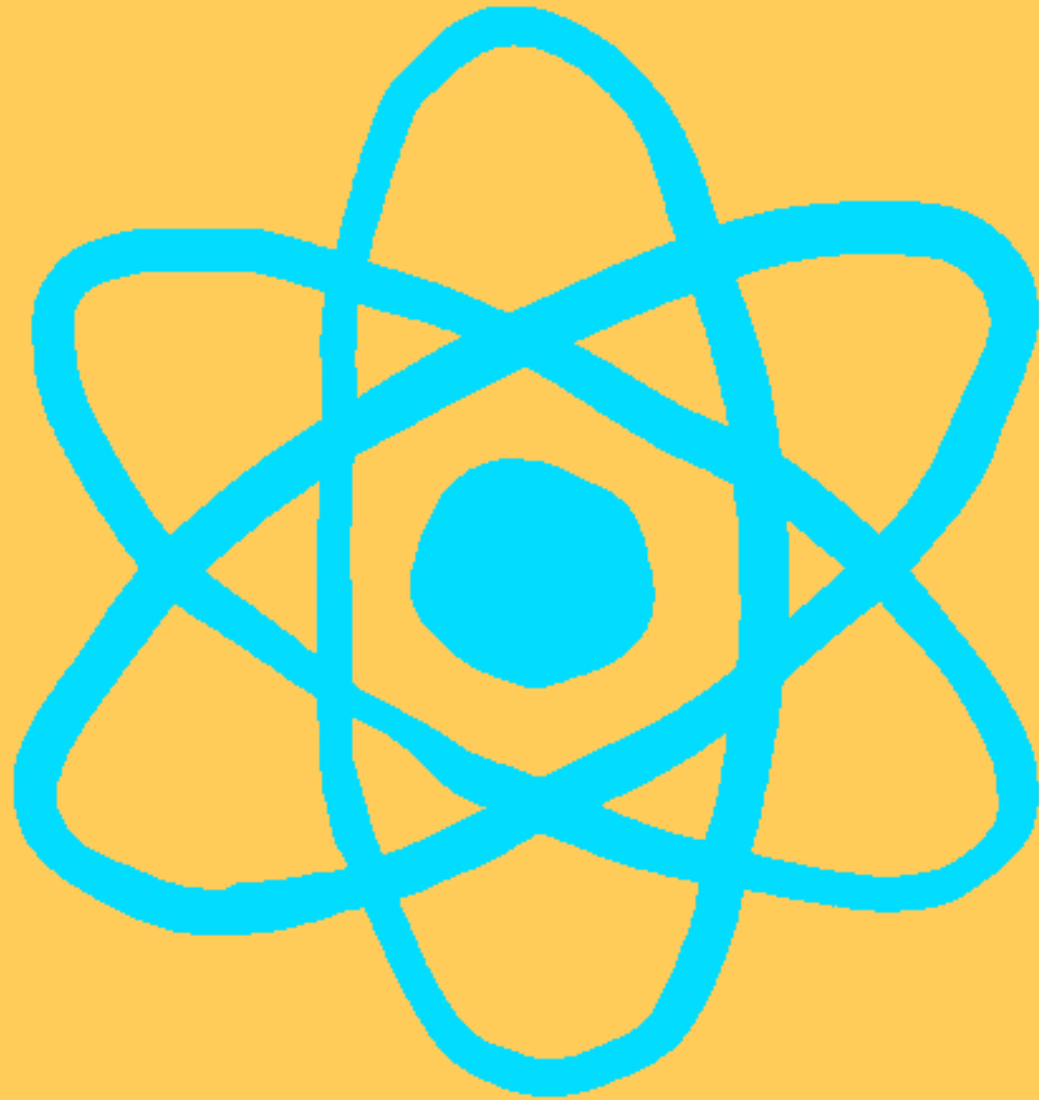


*FRONTEND. REACT. ЛЕКЦІЯ 5*

# HOOKS. PART 2. BASIC FORMS, FRAGMENTS, PORTALS.





## Створення користувацьких хуків

Хуки — це новинка в React 16.8. Вони дозволяють вам використовувати стан та інші можливості React без написання класу.

Створення власних хуків дозволить вам винести логіку компонента у функції, придатні для повторного використання.

Коли ми розглядали [використання хука ефекту](#), ми бачили

ВСТАНОВЛЕННЯ ▾

ОСНОВНІ ПОНЯТТЯ ▾

ПРОСУНУТІ ТЕМИ ▾

API-ДОВІДКА ▾

ХУКИ ▲

1. Знайомство з хуками
2. Огляд хуків
3. Використання хука ста
4. Використання хука ефекту
5. Правила хуків
- 6. Створення користувацьких хуків**
7. Довідка API хуків

# *REACT USE*

<https://github.com/streamich/react-use>



# ФРАГМЕНТИ

# ФРАГМЕНТИ

- В корні компоненту може бути лише 1 тег

# ФРАГМЕНТИ

- В корні компоненту може бути лише 1 тег
- Фрагмент використовуються там, де не потрібно мати тег в корені



# ФРАГМЕНТИ

- В корні компоненту може бути лише 1 тег
- Фрагмент використовуються там, де не потрібно мати тег в корені
- Неможливо додати події, стилі і т.д.

# ФРАГМЕНТИ. МОЖЛИВІ ФОРМИ ЗАПИСУ

## Скорочений запис

```
export function FragmentExample1() {  
  return (  
    < >  
      <span>Це перший фрагмент. </span>  
      <span>Скорочений запис</span>  
    < />  
  );  
}
```

# ФРАГМЕНТИ. МОЖЛИВІ ФОРМИ ЗАПИСУ

## React.Fragment

```
export function FragmentExample2() {  
  return (  
    <react.fragment>  
      <span>Це другий запис фрагмент. </span>  
      <span>React.Fragment запис</span>  
    </react.fragment>  
  );  
}
```

# ФРАГМЕНТИ. МОЖЛИВІ ФОРМИ ЗАПИСУ

## Fragment запис

```
export function FragmentExample3() {  
  return (  
    <fragment>  
      <span>Це третій фрагмент. </span>  
      <span>Fragment запис</span>  
    </fragment>  
  );  
}
```

# ПОРТАЛИ

# ПОРТАЛИ

- Версія після 16.3

# ПОРТАЛИ

- Версія після 16.3
- Можливість рендерити за межами кореневого елемента

# ПОРТАЛИ

- Версія після 16.3
- Можливість рендерити за межами кореневого елемента
- Працює, як звичайний реакт елемент



# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { createPortal } from "react-dom";
2
3 const modalRootEl = document.querySelector('#modal');
4
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   return open ? createPortal(
9     <div classname="modal-background" onclick="{onClose}">
10       <div classname="modal-content">
11         {props.children}
12       </div>
13     </div>,
14     modalRootEl
15   ) : null;
16 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```



# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```

# ПОРТАЛ

## СИНТАКСИС

```
1 import { Portal } from './components/Portal';
2
3 const [open, setOpen] = useState(false);
4 return (
5     <div>
6         <button onClick="{()}" ==""> setOpen(true)>Open</button>
7         <portal open="{open}" onClose="{()}"> setOpen(false)>>
8             <div>My first test modal</div>
9         </portal>
10    </div>
11 );
12 }
```

# WHO? WHO?

