

Refs. React Patterns. Final project.

Alen Murtić



01

Refs

02

React Portal

03

Controlled components

04

Final project



01

Refs

Refs

Refs

- React is declarative and abstracts DOM manipulations
- Refs are used to get reference on the underlying DOM element
 - Access DOM element properties (e.g. height, position, ...)
- [Official docs](#)
- Returns object with **current** key set to DOM object or **null**
- **useRef** hook - `const ref = useRef(null)`
- **createRef** method - `this.ref = createRef(null)`
- Refs are also used to store data which should not trigger re-renders



Refs example - refs.md



Refs

usePrevious

- Sometimes we need previous prop value in current component
 - In class components `componentDidUpdate` could access it
 - Use case example: animations
- We don't want to re-render on update of previous value
- Solution: `usePrevious` hook



Refs example - usePrevious.md



02



React Portal

React Portal

React Portal

- [Official docs](#)
- Render children outside of DOM hierarchy
 - Used with modals, dialogs, notifications, toasts
- Commonly used with Singleton pattern
- ReactDOM specific
- `ReactDOM.createPortal(child, container)`
- Event propagation and Context work as if it is still in the hierarchy



Portal example - portal.md




03



Controlled
components

Controlled components

Controlled components

- Components which are controlled by the parent via props
 - They receive **value** and **onChange** handler
- Often used in forms to allow the parent to always know current input values
 - Like you did in Input.tsx in your 5th homework
- Example: Tabs component
 - Controlled: receive current value and onChange/onClick as props
 - Uncontrolled: internally stores state and handles changed, only receives list of Tab instances and TabPanel instances as children (handled via context probably)
- Other React patterns: [High order components](#) and [Render prop](#)
 - Mostly made obsolete by custom hooks 

04



Final project

**Thank you for your
attention!**

