

# React – Hooks

Sladana Turudić - [sladjanaturudic@uns.ac.rs](mailto:sladjanaturudic@uns.ac.rs)

Teodora Ruvčeski - [teodoraruvceski@uns.ac.rs](mailto:teodoraruvceski@uns.ac.rs)



# Hooks

- Hooks su funkcije u Reactu koje omogućavaju korištenje stanja i drugih React funkcija bez pisanja klasa (koristimo za implementaciju logike i funkcionalnosti u komponentama).
- Najčešće korišćeni hook-ovi:
  - **useState**
    - Stanje je objekat koji sadrži informacije o određenoj komponenti. Obične JavaScript funkcije nemaju mogućnost pohranjivanja informacija. Kod unutar njih se izvršava i "nestaje" kada se izvršavanje završi.
    - Zahvaljujući stanju, React funkcionalne komponente mogu pohraniti informacije čak i nakon izvršenja.
  - **useEffect**
    - Odnosi se na nešto što se dešava nakon što se dogodi neka druga specifična stvar.



# useState Hook

- Prvo importujemo useState hook: `import { useState } from 'react'`
- Inicijalizacija stanja: `const [count, setCount] = useState(0)`
  - Ovdje dajemo ime varijable za stanje ( `count`) i ime funkcije koju ćemo koristiti za ažuriranje stanja ( `setCount`). Zatim postavljamo početnu vrijednost stanja (0), koja će biti vrijednost koja se učitava po defaultu svaki put kada se aplikacija pokrene.

Izvršava se tokom svakog renderovanja.

```
function decrementCount() {  
  setCount(count - 1);  
}
```

Moguće je da vrijednost count promjenljive nije najnovija.

```
1  import logo from './logo.svg';  
2  import './App.css';  
3  import { useState } from 'react';  
4  
5  function App() {  
6    const [count, setCount] = useState(4);  
7  
8    function decrementCount() {  
9      setCount((prevCount) => prevCount - 1);  
10   }  
11  
12   function incrementCount() {  
13     setCount((prevCount) => prevCount + 1);  
14   }  
15  
16   return (  
17     <>  
18       <button onClick={decrementCount}>-</button>  
19       <span>{count}</span>  
20       <button onClick={incrementCount}>+</button>  
21     </>  
22   );  
23 }  
24  
25 export default App;
```

```
const [count, setCount] = useState(() => {  
  return 4;  
});
```

Izvršava se samo jednom prilikom prvog renderovanja komponente.



# useState Hook

```
const [state, setState] = useState({ fontSize: 11, fontColor: "red" });  
  
function decrement() {  
  setState((prevState) => {  
    return { ...prevState, fontSize: prevState.fontSize - 1 };  
  });  
}
```



# useEffect Hook

- Prihvata dva argumenta. Prvo imamo *callback* koji izvršava našu funkciju, a zatim imamo niz koji se zove "*array of dependencies*". Ako ga ostavimo praznim povratni poziv će se izvršiti nakon što se komponenta renderuje.

```
const [count, setCount] = useState(4);

useEffect(() => {
  console.log("Count has changed!");
}, [count]);
```

```
useEffect(() => {
  console.log("This is my side effect");

  return () => {
    console.log("This is my clean up");
  };
});
```

- Cleaning Up Side Effects



# Termin 2 zadatak

- Kreirati formu za dodavanje novog studenta
- Student ima polja: ime, prezime, broj indeksa, naziv fakulteta
- Sva polja su obavezna, a ime i prezime imaju maks dužinu 30 karaktera
- Po kreiranju studenta korisnika prebaciti na URL `/success/studentId` gde je `studentId` broj indeksa studenta. Na tom URL-u treba da bude stranica koja ispisuje Uspešno ste kreirali studenta sa brojem indeksa `broj_indeksa_studenta`



THANK YOU

