11- React

React

- Open-source frontend JavaScript AJAXová knihovna pro návrh UI a UI komponent
- Vyvíjen Facebookem
- Existuje také framework React Native umožňující vývoj aplikací pro Android, iOS, Windows, ...

Komponenta

- Umožňuje rozdělit UI na části
- Jsou na sobě nezávislá a lze je používat opakovaně
- Lze zapsat jako třídu nebo funkci

DOM – Document Object Model

- Jazykově nezávislé rohraní umožňující přístup a aktualizaci prvků v dokumentu
- Reprezentuje dokument tak, aby program mohl měnit jeho strukturu a obsah
- Prohlížeč si stránku po jejím stažení převede na DOM
    - Všechny prvky stránky jsou díky tomu v paměti uloženy jako objekty a uspořádány do stromu
    - Díky DOM může JavaScript přistupovat ke všem HTML prvkům na stránce

Virtual DOM

- Používá ho např. React nebo Vue.js
- VDOM je abstraktnější, rychlejší a odlehčená reprezentace pravého DOMu
- Eliminuje zbytečné překreslování neměnných komponent
    1. Vytvořit VDOM s novým stavem aplikace
    2. Porovnat s předchozím VDOM
    3. V pravém DOM aktualizovat pouze ty komponenty, které se změnily

Událost

- Názvy událostí jsou camelCase, nikoliv lowercase

- Funkce se předává jako event handler, nikoliv string
- Nelze vracet false
- Není potřeba volat addEventListener

Props

- props = properties
- Objekt, který slouží jako vstup do komponenty
- Přenáší se v něm data, která poté lze použít v komponentě
- Props si lze "vyzvednout" dvěma způsoby

```
const Welcome = ({name, surname}) => { return (<><h1>Hello, {name} {surname}!</h1></>); }
const Welcome = (props) => { return (<><h1>Hello, {props.name} {props.surname}!</h1></>); }
```

State

- Každá class komponenta má vestavěný state
- Funkcionální komponenta může také mít svůj state díky háku useState
- Do statu se ukládají data (proměnné) komponenty
- Tato data "přežijí" překreslení komponenty – deklarace se znovu nezavolá (refresh stránky nepřežijí)
- Při změně těchto dat se komponenta překreslí

```
{/*const [<název proměnné>, <název metody pro změnu proměnné>] = useState(<iniciální hodnota>);*/}
const [name, setName] = useState("");
```

Hook

- Háky byly přidány ve verzi 16.8.0 (2019)
- Umožňují používat funkce Reactu bez použití třídy
    - Nefungují uvnitř tříd, pouze uvnitř React funkcí
- Musí se volat na nejvyšší úrovni, vždy ve stejném pořadí
    - Nelze je tedy volat ze smyček, podmínek apod.

useEffect

```
useEffect(() => {
    {/*Kód, který se má spustit*/}
}, []);
```

- Druhým parametrem háku useEffect je pole prvků – pokud jakýkoliv z nich změní svou hodnotu, je spuštěn kód uvnitř háku
    - Pokud je pole prázdné, slouží hák jako konstruktor

useContext

- Umožňuje mít společná globální data, která nemusíme všem komponentám tunelovat pomocí props
- V kombinaci s hákem useReducer lze mít i globální logiku

Propagace stavu

- Pokud v potomkovi chceme změnit hodnotu rodiče, předáme mu v props metodu na změnu této hodnoty