# 23 React

* Openc source syntax rozšíření pro JS
* React Native
* React Komponenty se oproti HTML skládají ve stromové struktuře Virtual DOM
* Kód je odsud kompilován do čistého JS pomocí React.createElement() a renderování v klasickém HTML DOM-u

## DOM – Document object model

* Jazykově nezávislé rozhraní 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

* React a Vue
* VDOM je abstraktnější, rychlejší a odlehčená verze pravého DOMu
* Struktura pravého DOMu u SPA může být komplexní a operace v ní tedy pomalé
* 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

## Komponenta

* Rozdělují UI aplikace do nezávislých znovu použitelných částí – enkapsulace
* Lze zapsat jako třídu / funkci
* Skládají se z elementů

Obsah obrázku text

Popis byl vytvořen automaticky

## Elementy

* Prvky HTML, které React renderuje
* Generují se metodou render()

## Props

* Properties
* Argumenty předávané react komponentům JSX atributy
* Předávání rodič>potomek



Obsah obrázku text

Popis byl vytvořen automaticky

## State

* Nejpoužívanější hook
* Get / set
* Get nám zobrazí nejaktuálnější podobu stavu
* useState – ukládání proměnných komponent
* Data jsou uchována i po překreslení komponenty
* Při změně těchto dat se může komponenta překreslit

## useEffect

* pokud se cokoliv v poli prvků změní – kód se uvnitř háku spustí
* prázdné pole – konstruktor
* Obsah obrázku text

  Popis byl vytvořen automaticky

## Context

* Globální úložiště dat aplikace
* Omezuje repetetivnost
* Pouze pro globální data
* Je možno napojit na LocalStorage, Session atd.
* Jeden stav je rozšířen jinak než rodič > potomek
* Nemusíme data všem komponentám tunelovat pomocí props
* S Reducerem může mít i svou logiku

## Eventy

* JSX atributy pro HTML/JS eventy
* Místo stringu můžeme předat funkci samotnou i s dalšími parametry z komponent
* Zápis camelCase
* Funkce se předává jako event handler
* Nelze vracet false
* Není potřeba volat addEventListener

Obsah obrázku text

Popis byl vytvořen automaticky

## Hooks

JS funkce, nevrací UI, zahákneme se s nimi do komponentů

Můžeme např detekovat update state-u nebo třeba API call-ů

useState, useEffect

Obsah obrázku text

Popis byl vytvořen automaticky

## Propagace stavu

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