

## 1 Úvod do Reactu

Napíšeme si první aplikace v Reactu a představíme si hlavní principy tohoto moderního frameworku.

◀ Nemá předchozí  
lekcí

Následující  
Podmíněné zobrazení ▶

Do této chvíle jsem naše aplikace psali v "čistém" (anglicky se též říká "vanilla") JavaScriptu. Postupy, které jsme se dosud naučili, jsou zcela legitimní samy o sobě a můžeme takto psát i větší aplikace. U aplikací, které jsou velmi interaktivní a často mění obsah stránky, však narazíme na různé problémy.

- Budeme muset náš kód zodpovědně strukturovat do komponent. Protože všechny komponenty se píšou velmi podobně, budeme mít spoustu kódu, který píšeme pořád dokola.
- Zařídit správnou komunikaci mezi velkým množstvím komponent začne být čím dál těžší.
- Při velkém množství komponent je poměrně těžké zajistit dobrý výkon aplikace.

Na tyto problémy tvrdě narazili vývojáři Facebooku, a proto vytvořili framework s názvem [React](#), který měl tyto problémy pomoci odstranit. Podobných frameworků existuje vícero. Mezi nejznámější vedle Reactu patří [Angular](#) a [Vue](#). Tato trojice je nejpoužívanějšími frameworky pro vývoj webů. V současné době je React stále s přehledem nejpopulárnější a také nejžádanější na trhu práce.

## První React aplikace

Ať už je to šikovná náhoda nebo dokonale promyšlený plán, velkou část reactových principů už jste v tomto kurzu používali, aniž byste o tom věděli. K vytvoření našeho prvního React projektu tak stačí pouze malinko upravit konfiguraci Webpacku a naučit se psát náš kód reactovým způsobem.

Základní Webpack projekt nastavený pro práci v Reactu si můžete vygenerovat pomocí `npm init kodim-app@latest`.

```
$ npm init kodim-app@latest my-app
```

Pozor na to, že tontokrát v příkazu chybí slovíčko `vanilla`. React projekt se používá stejným způsobem, jakým jsme u Webpack projektů zvyklí. Všimněte si však, že místo `index.js` zde máme soubor `index.jsx`. Příponu `.jsx` budeme používat u souborů, které obsahují reactový kód. Co tato přípona znamená, si povíme v následující části.

## JavaScript XML

Do této chvíle jsme viděli dva způsoby, jak v JavaScriptu vytvořit obsah stránky. Jedna z možností byla vytvořit řetězec obsahující HTML a ten pak vložit do stránky pomocí `innerHTML`.

```
const obsah = '<h1 class="title">Moje stránka</h1>';

const appElm = document.querySelector('#app');
appElm.innerHTML = obsah;
```

Toto řešení je relativně přímočaré, ale má nevýhody v tom, že je těžké udělat obsah interaktivní. Naučili jsme se proto druhý způsob, kdy místo řetězce vytvoříme DOM element.

```
const obsah = document.createElement('h1');
obsah.classList.add('title');
obsah.textContent = 'Moje stránka';

const appElm = document.querySelector('#app');
appElm.appendChild(obsah);
```

Toto řešení už je praktičtější, kód je však delší a méně přehledný. Zde proto s velkou fanfárou a na alegorickém voze přijíždí React se třetím způsobem, jak vytvářet obsah stránky.

```
const obsah = <h1 className="title">Moje stránka</h1>;
```

V tuto chvíli do proměnné *obsah* přímo bez obalení uvozovkami ukládáme něco, co vypadá podezřele jako HTML. Není to však přesně HTML, nýbrž takzvané JSX (*JavaScript XML*). Zápis JSX je velmi podobný jazyku HTML, jsou zde však drobné rozdíly. Například místo atributu `class` se v JSX používá atribut `className`.

JSX elementy jsou reactová obdoba DOM elementů, které jsme používali ve vanilla JavaScriptu. Jde o zcela nový typ hodnoty, a proto je můžeme stejně jako ostatní hodnoty ukládat do proměnných, předávat funkcím, vracet z funkcí, vkládat do polí i objektů a tak dále.

## Reactová aplikace

Pokud chceme náš JSX element vložit na stránku, musíme nejdříve v kódu založit reactovou aplikaci. To se dělá pomocí funkce `createRoot`. Této funkci předáme DOM element, uvnitř kterého se bude aplikace vykreslovat. Většinou to bývá `div`, kterému nastavíme `id` na `app`.

```
import { createRoot } from 'react-dom/client';

const appRoot = createRoot(document.querySelector('#app'));
```

Na objektu, který nám vrátí funkce `createRoot`, pak zavoláme funkci `render`, které už předáme JSX, které chceme vložit na stránku.

Aby nám React správně fungoval, musíme ještě na začátku každého souboru s příponou `.jsx` importovat balíček `react`. Celý soubor `index.jsx` tak bude vypadat takto.

```
import React from 'react';
import { createRoot } from 'react-dom/client';

const appRoot = createRoot(document.querySelector('#app'));
appRoot.render(<h1 className="title">Moje stránka</h1>);
```

Následující   
Obsah stránky v Reactu

# Kódím.cz

Verze 2.0.0-beta.7