React Router

Ondra Kučera, Tereza Vaňková, 30. 4. 2025

React girls

01

Opakování

Komponenty

- Komponenty jsou základními jednotkami, ze kterých stavíme aplikace
- Podobně jako DOM tvoří komponenty strom popisující obsah aplikace
- Komponenta je funkce
  - má jediný parametr datového typu objekt (*props*)
  - vrací JSX

Props

- Komponenta definuje svoje rozhraní pomocí tzv. *props*, které jí mohou či musejí být předány
  - rodičovská komponenta musí toto rozhraní dodržet
- Rodičovská komponenta předává pomocí *props* dětské komponentě svá data
  - *props* mohou být čísla, řetězce, pole, objekty, funkce, …
- Dítě rodiči data přímo předat nemůže
  - může ale definovat *prop* typu funkce, kterou rodič předá a dítě zavolá a tím rodiči předat data může

Stav

- Komponenta může udržovat svůj vnitřní stav (pomocí *useState*)
  - na základě něj může řídit, jak se vykreslí
  - na základě něj může měnit hodnotu *props* svých dětí a tím olivňovat, jak se vykreslí ony
- Pro každou stavovou proměnnou je potřeba dobře rozmyslet její výchozí hodnotu
  - a počítat s ní při vykreslení

Důsledky

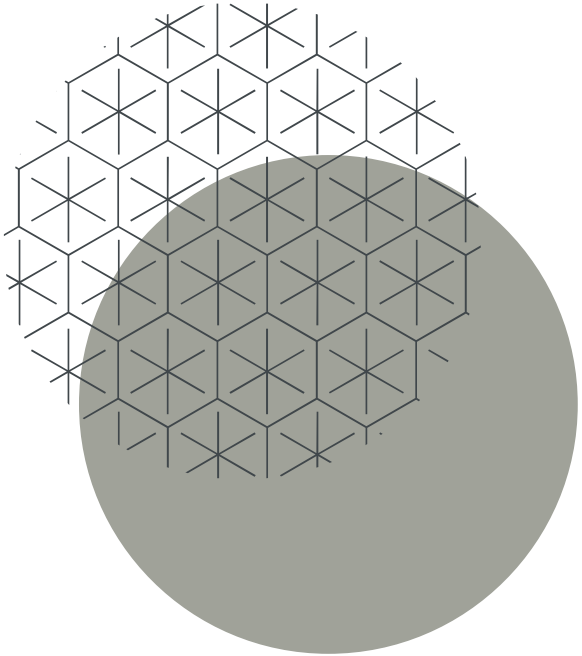
- Komponenta předává data svým dětem
  - často na základě svého stavu a jeho změn
  - děti mohou tato data či jejich část dále předávat svým dětem
- Dítě nemůže přímo informovat svého rodiče o změně svého stavu
  - je-li to potřeba, musí jednou z *props* dítěte být funkce, kterou dítě při změně svého stavu zavolá, aby rodiče informovalo

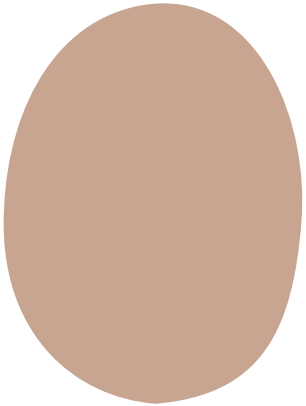
Důsledky

- Dítě nemůže přímo informovat sourozence (jiné dítě svého rodiče) o změně svého stavu
  - je-li to potřeba, musí jednou z *props* dítěte být funkce, kterou dítě při změně svého stavu zavolá, aby informovalo rodiče
  - rodič informaci může předat ostatním dětem pomocí *props*

02

npm





npm

- Nástroj pro práci se závislostmi (knihovnami) v javascriptových projektech
  - přesněji v projektech pro běhové prostředí Node (jsou i jiná javascriptová prostředí)
  - existují k němu i alternativy (Yarn, pNPM)

Vytvoření projektu

- Vytvoření prázdného projektu
  - npm init
- Použití některého z připravených generátorů
  - npm create vite@latest
- Klíčovým souborem projektu je *package.json*
  - popisuje základní informace o projektu (název, verzi, …)
  - zásadní informací je seznam závislostí projektu (knihoven, které projekt používá)

Instalace závislostí

- Závislosti projektu je nejprve potřeba stáhnout a nainstalovat
  - jsou instalovány do podsložky *node_modules*
- K instalaci závislostí slouží příkaz *npm install* (bez dalších parametrů)
  - příkaz si načte informace o závislostech ze souboru *package.json* a vytvoří (či doplní) obsah podsložky *node_modules*
  - některé generátory na začátku rovnou instalaci závislostí provedou samy
- Bez korektního obsahu *node_modules* nebudou věci správně fungovat!

Spuštění aplikace pro lokální vývoj

- npm run dev

Přidání nové závislosti

- npm install <název_knihovny>
  - zapíše informaci o závislosti do *package.json*
  - nainstaluje závislost do *node_modules*

03

Importování

Importování javascriptových identifikátorů

- Importování funkcí (popř. objektů, polí nebo jiných dat)
  - `import { usefulFunction1 } from "./useful-functions"`
- Speciálním případem je import komponent
  - `import { Component2 } from "./Component2"`

Importování CSS

- I CSS soubory typicky strukturujeme dle jednotlivých komponent
  - ke komponentě *Introduction* v souboru *Introduction.jsx* bude existovat soubor *Introduction.css*
  - použití v komponentě: import "./Introduction.css"
  - je vhodné v JSX komponenty *Introduction* zavést CSS třídu *Introduction* a selektory stylů odvozovat od ní

Importování obrázků

- Struktura složek uvnitř *src* nijak nesouvisí s výslednými URL aplikace
  - aplikace se už neskládá z jednotlivých statických HTML souborů, nýbrž z komponent dynamicky vykreslovaných JavaScriptem
- Podobně jako styly, i obrázky chceme umisťovat ke komponentám, ke kterým patří
  - jak ale vyplnit prop *src* komponenty *img*?

Importování obrázků

- I obrázky je potřeba importovat
  - `import logo from "./logo.svg"`
- Proměnnou *logo* potom můžeme předat jako prop *src*
  - `<img src={logo} alt="Logo ReactGirls" />`
- Při vykreslování naší komponenty se potom pro obrázek a jeho atribut *src* použije správná adresa

04

Rozdělení aplikace do samostatných stránek

Stránky v SPA

- Co v SPA znamená pojem stránka?
- Co v SPA znamená vytvořit odkaz z jedné stránky aplikace na jinou?

Příklad

Simulace odkazu pomocí stavu

Příklad

Použití knihovny React Router

Příklad: použití knihovny React Router

- Vytvoření projektu
  - npm create vite@latest
  - název projektu: react-router-example
- Nainstalování závislostí
  - npm install
- Přidání závislosti na knihovně *react-router-dom*
  - npm install react-router-dom
- Dokumentace knihovny React Router
  - https://reactrouter.com/en/main

Příklad: použití knihovny React Router

- Vytvoření komponent reprezentujících jednotlivé stránky
  - Introduction, Academies, ReactAcademy

Příklad: použití knihovny React Router

```
import { createBrowserRouter, RouterProvider } from "react-router-dom";
...

const router = createBrowserRouter([
   { path: "/", element: <Introduction /> },
   { path: "/akademie", element: <Academies /> },
   { path: "/react-akademie", element: <ReactAcademy /> },
]);

createRoot(document.getElementById("root")).render(
  <StrictMode>
     <RouterProvider router={router} />
  </StrictMode>
);
```

Příklad: použití knihovny React Router

```
import { Link } from "react-router-dom";
...

{/* uvnitř JSX: */}
      <p>
        Mimo jiného pořádáme řadu <Link to="/akademie">akademií.</Link>
      </p>
```

Cvičení

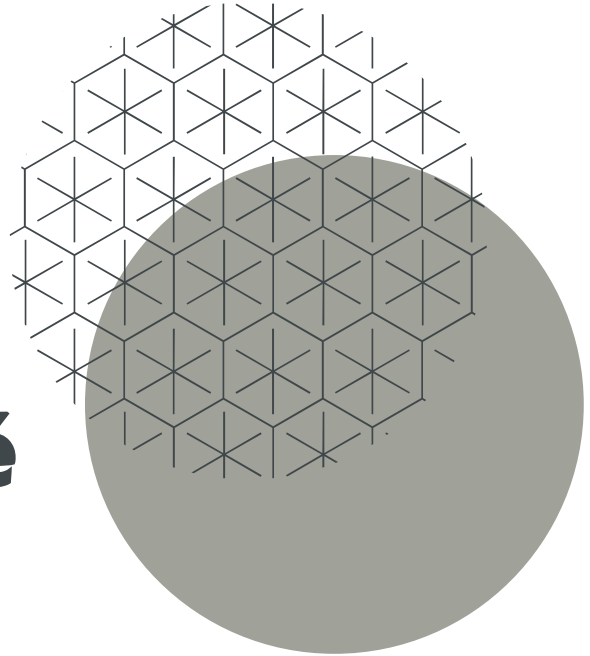
Čtvero ročních období

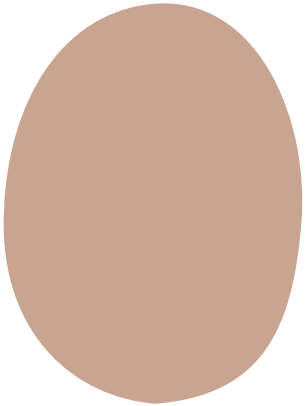
Cvičení: čtvero ročních období

- Vytvořte projekt *seasons*
- Přidejte závislost na knihovně *react-router-dom*
- Vytvořte následující strukturu stránek (definice routování probíhá v souboru *index.js*):
  - http://localhost:3000/
    - úvodní stránka se seznamem odkazů na popis jednotlivých ročních období
  - http://localhost:3000/spring
    - stránka s popisem jara
  - http://localhost:3000/summer
    - stránka s popisem léta
  - http://localhost:3000/autumn
    - stránka s popisem podzimu
  - http://localhost:3000/winter
    - stránka s popisem zimy

05

Parametry v cestě





Parametry v cestě

- Často je potřeba cesty v URL parametrizovat
- Příklad:
  - http://localhost:3000/employees/1
    - zobrazení detailu zaměstnance s ID 1
  - http://localhost:3000/employees/2
    - zobrazení detailu zaměstnance s ID 2
  - ...

Parametry v cestě

```
const router = createBrowserRouter([
    ...
    {
        path: "/employees/:id",
        element: <Employee />,
    },
]);
```

Parametry v cestě

```
import { useParams, Link } from "react-router-dom";

export const Employee = () => {
  const { id } = useParams(); // pozor: vždy řetězec!
   ...
};
```

Cvičení

Zaměstnanci

Cvičení: zaměstnanci

- Vytvořte projekt *employees*
- Přidejte závislost na knihovně *react-router-dom*
- Ve složce *src* vytvořte podsložku *data* a do ní nahrajte z Drivu soubory *employees.json* a *departments.json*
  - data z těchto souborů si následně můžete v kódu do proměnných uložit takto:
    - ```
      import employees from "./data/employees.json";
      import departments from "./data/departments.json";
      ```
      ```

Cvičení: zaměstnanci

- Vytvořte následující strukturu stránek
  - http://localhost:3000/
    - úvodní stránka se seznamem odkazů na detail jednotlivých zaměstnanců
  - http://localhost:3000/employees/:id
    - stránka s detailem konkrétního zaměstnance
    - detail zaměstnance obsahuje jeho jméno, věk a název oddělení, do kterého patří

06

Komponenta Outlet

Příklad

Použití komponenty Outlet

07

Shrnutí

React Router

- Definice adres v aplikaci pomocí funkce createBrowserRouter
- Odkazy uvnitř aplikace pomocí komponenty Link
- Získání parametru z adresy pomocí hooku useParams

Díky za pozornost!