

## 6 Komunikace dítě rodič

Představíme si základní scénáře komunikace mezi komponentami a jejich použití v praxi.

◀ Předchozí  
**Formulářové prvky, efekty**

Následující ▶  
**Komunikace mezi sourozenci**

### 1 Mixér barev

Cílem projektu je vytvořit reactovou aplikaci, která umožňuje namíchat si barvu ze složek RGB tak, jak jste na to zvyklí z různých grafických programů. Na začátku máte připravenou funkční aplikaci v čistém JavaScriptu. Vaším úkolem bude tuto aplikaci přepsat do Reactu.

1. Vzor funkční aplikace v čistém JavaScriptu je v repozitáři [projekt-mixer-barev-js](https://github.com/Czechitas-podklady-WEB/projekt-mixer-barev-js). **Nepoužívejte repozitář jako šablonu**, nebudeme v něm dělat žádné úpravy. Repozitář si pouze naklonujte příkazem `git clone https://github.com/Czechitas-podklady-WEB/projekt-mixer-barev-js.git`. Otevřete si projekt ve VS Code. Stránka je napsána v čistém JavaScriptu bez Webpacku, stačí ji tedy spustit pomocí `npx serve`. Vyzkoušejte si, jak aplikace funguje.
2. Budeme chtít tuto aplikaci přepsat do Reactu. Za tímto účelem si založte vlastní prázdný repozitář na vašem GitHubu. Naklonujte si jej k sobě do počítače a vytvořte v něm základ reactové aplikace pomocí

```
npm init kodim-app@latest .
```

3. Vytvořte hlavní komponentu `App`, která zobrazuje celý obsah stránky včetně stylů a obrázků podle zadání. Stránka zatím nebude funkční, protože původní JavaScriptový kód nám v Reactu fungovat nebude.
4. Vytvořte nejprve komponentu `ColorSlider` zobrazující jeden posuvník. Přesuňte do složky s komponentou správné CSS styly. Komponenta `ColorSlider` bude mít dvě props: `baseColor` a `colorName`. Prop `baseColor`

udává, jakou posuvník ovládá barvu, tedy `red`, `green` nebo `blue`. Prop `colorName` udává zobrazený název barvy.

5. Použijte komponentu `ColorSlider` k zobrazení všech tří posuvníků uvnitř `App`.
6. Opět uvnitř komponenty `App` vytvořte tři stavy `redValue`, `greenValue`, `blueValue`. Každý reprezentuje hodnotu jedné z barevných složek. Použijte tyto hodnoty k vytvoření finální barvy uvnitř prvku `.color-box`. K tomu budete potřebovat vědět jak se v Reactu nastavují inline styly. Hezký článek najdete například na [w3schools](https://www.w3schools.com/css/css-inline.asp). Vyzkoušejte, si, že když stavům nastavíte nějaké výchozí hodnoty, bude na stránce vidět správně namíchaná barva.
7. Nyní budeme chtít propojit hodnoty posuvníků se těmito stavy aby se barva měnila v reakci na vstup od uživatele. Budeme tedy potřebovat komunikaci dítě -> rodič. `ColorSlider` (dítě) musí být schopen sdělit komponentě `App` (rodič), že uživatel změnil hodnotu na posuvníku.
8. V komponentě `ColorSlider` přidejte novou prop s názvem `onValueChange`. Bude to callback s jedním parametrem `value`. Tuto funkci komponenta zavolá pokaždé, když se změní hodnota posuvníku. Volání této funkce je způsob, jak předat hodnotu posuvníku nahoru do `App`. Funkci tedy zavolejte v reakci na událost `onChange` na posuvníku a předejte jí aktuální hodnotu v posuvníku.
9. Jako poslední krok stačí každé komponentě `ColorSlider` předat funkci pro změnu stavu skrze prop `onValueChange`. `ColorSlider` pro červenou barvu tak v prop `onValueChange` dostane funkci `setRedValue` atd.
10. Pokud jste všechno propojili správně, měla by vaše aplikace nyní fungovat.

# Kódím.cz

Verze 2.0.0-beta.7