

# React

**React** změnil způsob, jakým vývojáři přemýšlejí o tvorbě webů. Namísto manipulace s celou stránkou jako v [jQuery](#), React rozděluje uživatelské rozhraní na malé, samostatné a znovupoužitelné kousky zvané **komponenty**.

## 1. Klíčové koncepty

### Komponenty

Každá část webu (tlačítko, formulář, navigační lišta) je v Reactu samostatná komponenta. Tyto komponenty lze do sebe vkládat, kombinovat je a používat opakovaně napříč celou aplikací.

### Virtuální DOM (Virtual DOM)

Jedná se o největší technickou inovaci Reactu. Místo toho, aby React při každé změně překresloval celou skutečnou webovou stránku (což je pomalé), vytvoří si v paměti její lehkou kopii.

1. Dojde ke změně dat.
2. React vypočítá rozdíl mezi starým a novým virtuálním DOMem (tzv. **diffing**).
3. Pouze ty části, které se skutečně změnily, jsou následně aktualizovány v prohlížeči.

### JSX (JavaScript XML)

React používá speciální syntaxi, která umožňuje psát HTML kód přímo uvnitř JavaScriptu. Díky tomu je kód velmi přehledný a snadno se v něm pracuje s logikou.

### Jednosměrný tok dat (One-way Data Flow)

Data v Reactu proudí vždy shora dolů (od rodičovských komponent k potomkům). To usnadňuje hledání chyb a dělá aplikaci předvídatelnější.

## 2. Proč zvolit React?

- **Rychlost:** Díky virtuálnímu DOMu jsou aplikace extrémně svižné i při velkém množství dat.
- **Znovupoužitelnost:** Jednou napsané komponenty (např. stylové tlačítko) můžete použít v deseti různých projektech.
- **Ekosystém:** Existuje obrovské množství doplňků, komunitní podpory a hotových řešení.
- **React Native:** Znalost Reactu vám umožní vyvíjet i nativní mobilní aplikace pro iOS a Android.

### 3. Hooks - Srdce moderního Reactu

Od verze 16.8 používá React tzv. **Hooks** (háčky), které umožňují používat stav a další funkce bez psaní složitých tříd. Nejpoužívanější jsou:

- **useState:** Pro uchování dat, která se mění (např. text v políčku).
- **useEffect:** Pro provádění „vedlejších účinků“, jako je načítání dat z API po načtení stránky.

### 4. Ukázka kódu (Jednoduché počítadlo)

```
import React, { useState } from 'react';

function Pocitadlo() {
  const [pocet, setPocet] = useState(0);

  return (
    <div>
      <p>Klikli jste {pocet} krát</p>
      <button onClick={() => setPocet(pocet + 1)}>
        Klikni mě
      </button>
    </div>
  );
}
```

### 5. React vs. Angular vs. Vue

| Vlastnost   | React                         | Angular               | Vue                   |
|-------------|-------------------------------|-----------------------|-----------------------|
| Výrobce     | Meta (Facebook)               | Google                | Komunita (Evan You)   |
| Typ         | Knihovna                      | Robustní Framework    | Progresivní Framework |
| Učení       | Středně těžké                 | Velmi těžké           | Snadné                |
| Flexibilita | Vysoká (vybíráte si knihovny) | Nízká (vše je v ceně) | Střední               |

**Zajímavost:** React původně vyvolal vlnu kritiky, protože míchal HTML a JS (v JSX), což se tehdy považovalo za „špatnou praxi“. Čas ale ukázal, že tento přístup je pro vývoj uživatelských rozhraní mnohem efektivnější, a dnes ho napodobuje většina konkurence.

[Zpět na Architekturu](#)

From:  
<https://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:  
<https://serviceit.cz/doku.php?id=react>

Last update: 2025/12/31 17:24



