

## 3 Zobrazování seznamů

Ukážeme si, jak v React komponentách zobrazovat seznamy.

[◀ Předchozí](#)  
**Podmíněné zobrazení**

**Následující** [▶](#)  
**Události, stav**

## Složitější komponenty

Z praktického hlediska je náš ilustrativní příklad výše stále hodně jednoduchý. V praxi míváme mnohem obsáhlejší data i mnohem složitější komponenty. Vezměme například data pro nám již známý nákupní seznam.

```
const list = [  
  { product: 'mrkev', amount: '3ks', done: false },  
  { product: 'paprika', amount: '2ks', done: true },  
  { product: 'cibule', amount: '2ks', done: false },  
  { product: 'čínské zelí', amount: '1ks', done: true },  
  { product: 'arašídý', amount: '250g', done: false },  
  { product: 'sojová omáčka', amount: '1ks', done: false },  
];
```

Pokud tento seznam budeme chtít zobrazit, budeme potřebovat složitější JSX strukturu. Například jako v následující ukázce.

```
const App = () => (  
  <>  
    <h1>Nákupní seznam</h1>  
    <div className="shopping-list">  
      {list.map((item) => {  
        const tickClass = item.done  
          ? 'item__done item__done--tick'  
          : 'item__done';  

```

```
    return (  
      <li className="item">  
        <div className="item__product">{item.product}</div>  
        <div className="item__amount">{item.amount}</div>  
        <div className={tickClass}></div>  
      </li>  
    );  
  }  
}</div>  
</>  
);
```

Všimněte si, že funkce, která vyrábí JSX pro jednotlivé položky našeho seznamu, je už o kus obsáhlejší. Dokonce o tolik, že už nejde napsat zkráceným způsobem a musíme použít složené závorky a `return`. Tato ukázka opět poslouží jako dobré cvičení na pozornost ohledně závorek.

V praxi však často nastane situace, že funkce použitá uvnitř `map` je tak složitá, že je těžké se v kódu orientovat. Náš poslední příklad už je také trochu na hraně. V takovém případě si vzpomeneme na předchozí lekci, kde jsme probírali rozdělování jedné velké komponenty na menší celky. Rozhodně se nám vyplatí vytvořit si pro zobrazování jednotlivých prvků seznamu komponentu, například takto.

```
const ShoppingItem = ({ product, amount, done }) => {  
  const tickClass = done ? 'item__done item__done--tick' : 'item__done';  
  return (  
    <li className="item">  
      <div className="item__name">{name}</div>  
      <div className="item__amount">{amount}</div>  
      <div className={tickClass}></div>  
    </li>  
  );  
};
```

Možná jste ji už někde viděli. Nyní stačí ji použít v komponentě `App`.

```
const App = () => (  
  <>  
    <h1>Nákupní seznam</h1>  
    <ul className="shopping-list">  
      {list.map((item) => (  
        <ShoppingItem  
          product={item.product}  
          amount={item.amount}  
          done={item.done}  
        />  
      ))}  
    </ul>  
  </>  
);
```

Takto je náš kód mnohem čitelnější. Toto je jeden z hlavních důvodů, proč rozdělujeme naše aplikace na komponenty. Lépe se tak dokážeme v kódu orientovat nejen my, ale také naši kolegové, kteří pracují na stejném projektu a musí číst náš kód.

 Předchozí  
**Seznamy v JSX**

Následující   
**Práce s key prop**

# Kódím.cz

Verze 2.0.0-beta.7