

2 Podmíněné zobrazení

Naučíme se, jak sestavovat CSS třídy a jak se rozhodnout, kterou část JSX chceme zobrazit.

[◀ Předchozí](#)
[Úvod do Reactu](#)

[Následující >](#)
[Zobrazování seznamů](#)

Podmíněné zobrazení

Pro účely následujícího výkladu si představme jednoduchou komponentu `Product`, která zobrazuje jednoduchý produkt v nějakém e-shopu.

```
const Product = (props) => {  
  const { name, img } = props;  
  
  return (  
    <div className="product">  
      <div className="product__name">{name}</div>  
      <img className="product__img" src={img} />  
      <button>Vložit do košíku</button>  
    </div>  
  );  
};
```

Tato komponenta hezky ukazuje, kam všude můžeme do JSX vložit JavaScriptovou hodnotu pomocí složených závorek. Říkáme tomu "otevírání JavaScriptového okna".

Za prvé můžeme JavaScriptové okno otevřít, když předáváme hodnotu nějaké prop.

```
<img className="product__img" src={img} />
```

Zároveň také můžeme okno otevřít, když chceme vytvořit obsah nějakého elementu.

```
<div className="product__name">{name}</div>
```

Obsah elementu však může být mnohem složitější než pouze hodnota uvnitř jedné proměnné. Představme si, že pokud je produkt nedostupný, chceme místo tlačítka zobrazit element se zprávou. Přidáme tedy do komponenty novou prop `isAvailable` a použijeme náš oblíbený podmíněný operátor.

```
const Product = (props) => {  
  const { name, img, isAvailable } = props;  
  
  return (  
    <div className="product">  
      <div className="product__name">{name}</div>  
      <img className="product__img" src={img} />  
      {isAvailable ? <button>Vložit do košíku</button> : <div>Nedostupné<  
    </div>  
    );  
};
```

Všimněte si, že na místě, kde chceme provést rozhodnutí, otevřeme JavaScriptové okno, abychom mohli použít náš podmíněný operátor. Ten pak dle hodnoty v prop `isAvailable` vrátí příslušný kus JSX.

Pokud v podmíněném operátoru pracujeme s obsáhlejšími JSX, často se nám celý výraz nevejde na jeden řádek. Pak je ve zvyku pomoci si v ternárním operátoru kulatými závorkami.

```
const Product = (props) => {  
  const { name, img, isAvailable } = props;  
  
  return (  
    <div className="product">  
      <div className="product__name">{name}</div>  
      <img className="product__img" src={img} />  
    </div>  
  );  
};
```

```
    {isAvailable ? (  
      <button>Vložit do košíku</button>  
    ) : (  
      <div className="not-available">Nedostupné</div>  
    )}  
  </div>  
);  
};
```

Občas se nám stane, že v jednom z případů podmínky nechceme zobrazit nic. Dejme tomu, že nechceme uživateli zobrazovat zprávu o nedostupnosti, ale chceme pouze vynechat objednávací tlačítko. V takovém případě můžeme vrátit hodnotu `null`, která pro React znamená, že nemá zobrazit nic.

```
const Product = (props) => {  
  const { name, img, isAvailable } = props;  
  
  return (  
    <div className="product">  
      <div className="product__name">{props.name}</div>  
      <img className="product__img" src={props.img} />  
      {isAvailable ? <button>Vložit do košíku</button> : null}  
    </div>  
  );  
};
```

◀ Předchozí
Cvičení: Podmíněné výrazy

Následující >
Rozdělování komponent

Verze 2.0.0-beta.7