


# React pre začiatočníkov

## III. Formuláre

---

Ing. Slavomír Kožár

20. 3. 2022



# Obsah

- HTML komponenty formulářů
- (Ne) Kontrolované komponenty
- Študentský formulář

# Template

<https://slavokozar.github.io/react-pre-zaciatocnikov-form-student/>

```
$ git clone https://github.com/slavokozar/react-pre-zaciatocnikov-form-student.git
```

```
$ cd react-pre-zaciatocnikov-form-student
```

```
$ npm install
```

```
$ npm start
```

# Label

Určený pre popis vstupu.

Pri správnom nastavení `htmlFor` a `id` aktivuje vstup ( `focus` )

```
<label htmlFor="name">Meno</label>
```

```
<input id="name" />
```

# Input

`<input type="button">`

`<input type="checkbox">`

`<input type="color">`

`<input type="date">`

`<input type="datetime-local">`

`<input type="email">`

`<input type="file">`

`<input type="hidden">`

`<input type="image">`

`<input type="month">`

`<input type="number">`

`<input type="radio">`

`<input type="password">`

`<input type="range">`

`<input type="reset">`

`<input type="search">`

`<input type="submit">`

`<input type="tel">`

**`<input type="text">`**

`<input type="time">`

`<input type="url">`

`<input type="week">`

# Input

Meno

```
<input type="text">
```

# Select

Univerzita

Univerzita

- ✓ Prosím vyberte univerzitu
- ČVUT
- Univerzita Karlova
- Česká Zemědělská Univerzita
- Vysoká Škola Ekonomická

```
<select>
```

```
  <option value="">Prosím...</option>
```

```
  <option value="CVUT">ČVUT</option>
```

```
  <option value="CZU">ČZU</option>
```

```
  ...
```

```
</select>
```

# Radio

Typ štúdia ☐ Bakalárske ☐ Magisterské ☐ Doktorské

☐ Bakalárska  
☐ Magisterská  
☐ Doktorská

```
<input type="radio" name="stage" id="bachelor" value="bachelor"/>
<label htmlFor="bachelor">Bakalárska</label>
<input type="radio" name="stage" id="master" value="master"/>
<label htmlFor="master">Magisterská</label>
...
```



# Checkbox

☐ Distancne studium

☒ Distancne studium

```
<input type="checkbox" id="distance"/>
```

```
<label htmlFor="distance">Distancne studium</label>
```

# Textarea

Správa (max 200 znakov)



```
<label htmlFor="message">Správa</label>  
<textarea id="message" cols="30" rows="10"/>
```

# Nekontrolované komponenty

Všetky běžné vstupy tvoří nekontrolované komponenty.

**Ich stav si řadí prohlídač = React ich nemá pod kontrolou**

# Kontrolované komponenty

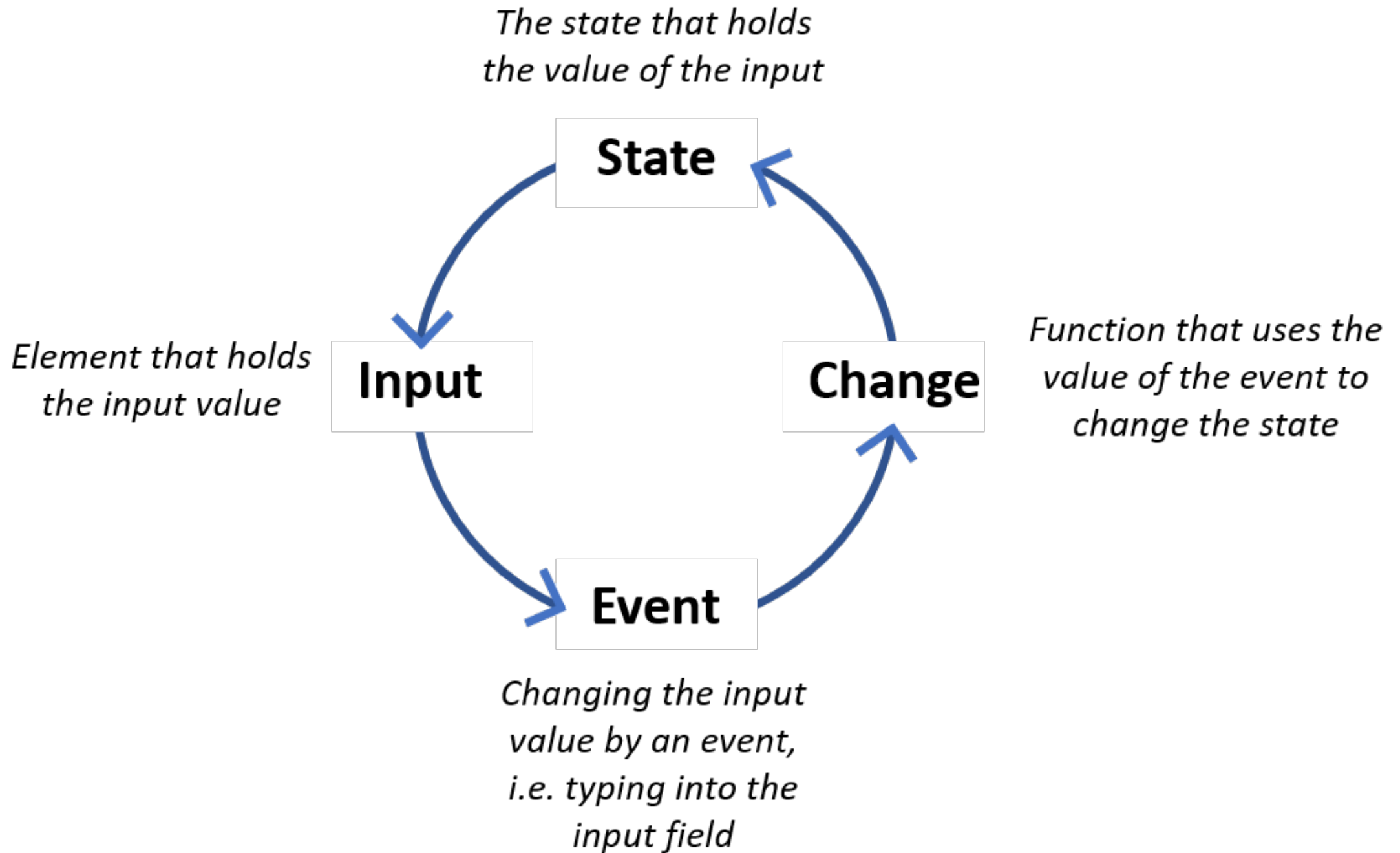
Stav vstupu je plne pod kontrolou reactu.

Nazývané aj **two way data binding** (názov pochádza z Angular.js).

Stav komponenty je **uložený v stave rodičovskej komponenty**.

# Kontrolované komponenty

```
function RodičovskáKomponenta () {  
  return (  
    <>  
      <label>Meno</label>  
      <input type="text" />      kontrolovaná komponenta  
    </>  
  )  
}
```



# State

- Kontrolovaná komponenta musí mať alokovaný priestor v stave rodičovskej komponenty.

```
const [name, setName] = useState('');
```

# Input

- Kontrolovaná komponenta musí zobrazovať hodnotu uloženú v rodičovskej komponente

```
const [name, setName] = useState('');
```

```
<input type="text" value={name} />
```



# Event

- React musí reagovať na zmenu kontrolovanej komponenty

```
const [name, setName] = useState('');
```

```
function handleNameChange (e) {  
}
```

```
<input type="text" value={name}  
  onChange={handleNameChange}  
/>
```

# Event

- React musí reagovať na menu kontrolovanej komponenty
- použitie arrow funkcie

```
const [name, setName] = useState('');
```

```
<input type="text" value={name}  
  onChange={ (e) => {} }  
/>
```

# Event Objekt

- Objekt automaticky vytvorený v momente udalosti
- Popisuje okolnosti udalosti

```
function handleNameChange (e) {  
    console.log(e) ;  
}
```

```
<input type="text" value={name}  
    onChange={handleNameChange}  
/>
```

# Change

- Zmena kontrolovanej komponenty musí zmeniť stav rodičovskej komponenty

```
const [name, setName] = useState('');
```

```
function handleNameChange (e) {  
  setName(e.target.value);  
}
```

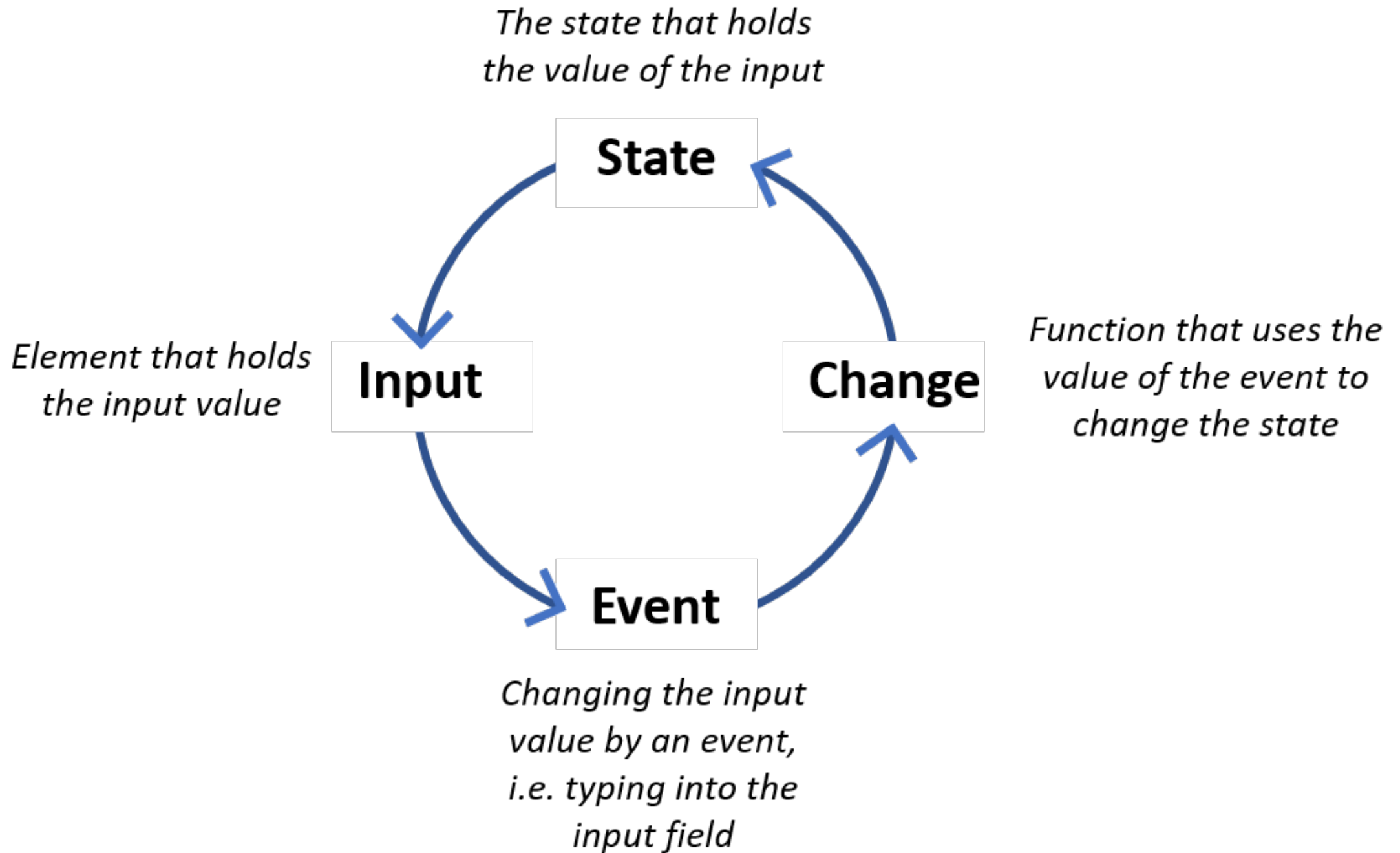
```
<input type="text" value={name}  
  onChange={handleNameChange}  
/>
```

# Change

- Zmena kontrolovanej komponenty musí zmeniť stav rodičovskej komponenty
- použitie arrow funkcie

```
const [name, setName] = useState('');
```

```
<input type="text" value={name}  
  onChange={ (e) => setName(e.target.value) }  
>
```



# Select

- Select sa používa rovnako ako bežný Input

```
const [faculty, setFaculty] = useState('')
```

```
<select  
  value={faculty}  
  onChange={ (e) => setFaculty(e.target.value) }  
>  
  ...  
</select>
```

# Checkbox

- Stav je boolean premenná – nepoužíva atribút **checked** nie value

```
const [distance, setDistance] = useState(false)
```

```
<input type="checkbox"  
      checked={distance}  
      onChange={ (e) => setDistance(e.target.checked) }  
/>
```



# Radio

- Z užívateľského aj stavového pohľadu rovnaký ako Select

```
const [stage, setStage] = useState('')
```

```
<input type="radio" value="bachelor"  
  checked={stage === 'bachelor'}  
  onChange={ (e) => {  
    setStage(e.target.checked ? 'bachelor' : stage)  
  }}  
>
```

# Výber fakulty


React má zaujímavú vlastnosť – dokáže vykresliť pole komponent.

Pravý význam tejto vlastnosti si ukážeme neskôr.

```
const faculties = {  
  "UK": [  
    <option value="Katolícka ..." >Katolícka ...</option>,  
    <option value="Evangelická ..." >Evangelická ...</option>,  
    ...  
  ]  
}
```

**Congratulations!**



 homm.fun

meme-arsenal.ru

# ĎAKUJEME !

---

**Ing. Slavomír Kožár**

slavomir.kozar@innovis.sk

[www.innovis.sk](http://www.innovis.sk)

