

# Vue.js

Components and emits

POWERED BY:



POWERED BY:



1. Components
2. Emits

# Components

- ```
Vue.component('button-counter',{  
  data: function () {  
    return {  
      count: 0  
    }  
  },  
  template: `  
    <button v-on:click="count++">  
      You clicked me {{ count }} times.</button>  
  `;  
});
```
- ```
<div id="app">  
  <button-counter></button-counter>  
</div>
```

# Components

- reusable functional templates
- data mora biti funkcija
- svaka komponenta mora imati single root element
- separation of concerns
- single-file components (.vue files) za velike projekte
- razlika pristupa - sve u jednom fajlu, protiv 3 vrste fajla kojima se komunikacija isprepliće (MVC)

## Components - tips

- registracije komponenti su globalne
- lokalne komponente deklariraju se kao JS varijable (objects)
- koristite single-file komponente gotovo uvijek

# Components

- ime single-file komponente mora se eksportati
- komponente moraju biti eksportane i proslijeđene putem components svojstva u objekt instance
- `data: { ... },`  
`components: { Home, About, Contact }`
- komponente se mogu koristiti unutar drugih komponenti  
`<template id="home">`  
`<About />`  
`</template>`

## EXAMPLE

## Components

- parent komponente mogu proslijediti podatke u child komponente pomoću props-a:

```
Vue.component('item', {  
  props: [ 'name' ],  
  template: '<h1>{{ name }}</h1>'  
})
```

- `<item name="My item"></item>`
- props se mogu proslijediti dinamički putem `v-bind`
- props-i bez vrijednosti imaju defaultnu boolean vrijednost - `true`

## Prop validations

```
→ props: {  
  propA: Number, propB: [String, Number],  
  propC: { type: String, required: true },  
  propD: { type: Number, default: 123 },  
  propE: { type: Object, default: function () {  
    return { name: "John Doe" } },  
  propF: { validator: function (value) {  
    return [ 'success', 'ok' ].indexOf(value) !== -1;  
  }}  
}
```



## Zadatak 1.

- Napravite Vue aplikaciju koja će imati glavnu Home komponentu. Home komponenta treba imati definiran niz objekata koji sadrže ID i naziv i predstavljaju neke proizvode u webshopu.

Proizvod je definiran komponentom Product koja ispisuje naziv proizvoda. Validirajte naziv kao obavezan property tipa string. [CODE](#)

## Computed properties

- da bi izbjegli lošu čitljivost i složene izraze, posebice ako se svojstvo ne mijenja, koristite computed properties:
- ```
{{ reversedMessage }}
```

```
data: {  
  message: 'Hello'  
},  
computed: {  
  reversedMessage() {  
    return this.message.split('').reverse().join(''  
  }  
}
```

## Emits

- slanje poruka od childa prema parentu
- `<button v-on:click="$emit('send-message')">`  
Send message  
`</button>`
- `<parent v-on:send-message="readMessage"></parent>`
- `<button v-on:click="$emit('send-message', 123)">`  
Send message  
`</button>`
- `methods: { readMessage: function (data) {} }`
- payload je dostupan i preko `$event` variable
- `<elem v-bind:title.sync="doc.title"></elem>`

## Using v-model on custom components

- ```
Vue.component('custom-input', {  
  props: ['value'],  
  template: `  
    <input v-bind:value="value"  
      v-on:input="$emit('input', $event.target.value)">  
  `,  
})
```
- ```
<custom-input v-model="searchText"></custom-input>
```

## Slots

- slots koristimo da proslijedimo stari sadržaj komponenti
- `<alert-box>`  
Something bad happened.  
`</alert-box>`
- `Vue.component('alert-box', {`  
  `template: ``  
    `<div class="demo-alert-box">`  
      `<strong>Error!</strong>`  
      `<slot></slot>`  
    `</div>`  
  `,`  
  `})`

## Dynamic and async components

- dinamička promjena komponente (e.g. tabs)
- `<component v-bind:is="currentTabComponent"></component>`
- value proslijeđen preko "is" ime je komponente koja će se izrenderirati
- ```
Vue.component('async', function (resolve, reject) {  
  setTimeout(function () {  
    resolve({  
      template: '<div>I am async!</div>'  
    })  
  }, 1000)  
})
```

## Zadatak 2.

- Napravite Vue aplikaciju koja će sadržati komponente Books i Book. Books sadrži definiciju knjiga, a Book sadrži pojedinačnu knjigu. Pojedinačne knjige trebaju se pojaviti 2s nakon klika na dugme “Show books”. Naziv knjige prikažite boldirano koristeći slotove. Cijenu knjige proslijedite iz parenta. Dodajte dugme u Book kojim će se knjiga pobrisati iz parent komponente. [CODE](#)

## Za vježbu

- Napravite Vue aplikaciju koja će sadržati komponente Shop i Item. U Shopu se nalazi više proizvoljnih itema. Klikom na dugme “Lower price” u Item komponenti, dohvatite nasumičan popust iz Shop komponente, u iznosu od 5 do 25 posto, i smanjite prikazanu cijenu. Svaki Item mora imati naziv i cijenu.



POWERED BY:



THANK YOU  
for attention