Τεχνολογίες Εφαρμογών Ιστού

VueJS

Konstantinos Tserpes

Introduction

- Vue (pronounced /vjuː/, like view)
- Framework for building user interfaces
- Links data and DOM

Development environment

- Include
  - Dev version: <script src="https://vuejs.org/js/vue.js"></script>
  - Production version: <script src="https://vuejs.org/js/vue.min.js"></script>

Declarative rendering

- Render data to the DOM using straightforward template syntax
- Vue mounts a Vue instance on an HTML element

```
<div id="app">
  {{ message }}
</div>
```

HTML element

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

Vue instance

Mount point

- The “el” property defines the mount point in the Vue instance
- Mount by id: use ‘#’, e.g. el: ‘#main’
- Mount by class: use ‘.’, e.g. el: ’.bar’

Directives

- v-bind: bind element attributes
- v-if: conditional
- v-for: loop
- v-on:{event name}: handling events
- v-model: makes two-way binding between form input and app state
- v-html: equivalent to “mustaches” {{}}

v-bind example

```
div id="app-2">
  <span v-bind:title="message">
    Hover your mouse over me for a few seconds
    to see my dynamically bound title!
  </span>
</div>
var app2 = new Vue({
  el: '#app-2',
  data: {
    message: 'You loaded this page on ' + new Date().toLocaleString()
  }
})
```

v-if example

```
<div id="app-3">
  <span v-if="seen">Now you see me</span>
</div>
var app3 = new Vue({
  el: '#app-3',
  data: {
    seen: true
  }
})
```

v-for example

```
<div id="app-4">
  <ol>
    <li v-for="todo in todos">
      {{ todo.text }}
    </li>
  </ol>
</div>
var app4 = new Vue({
  el: '#app-4',
  data: {
    todos: [
      { text: 'Learn JavaScript' },
      { text: 'Learn Vue' },
      { text: 'Build something awesome' }
    ]
  }
})
```

v-on:event example

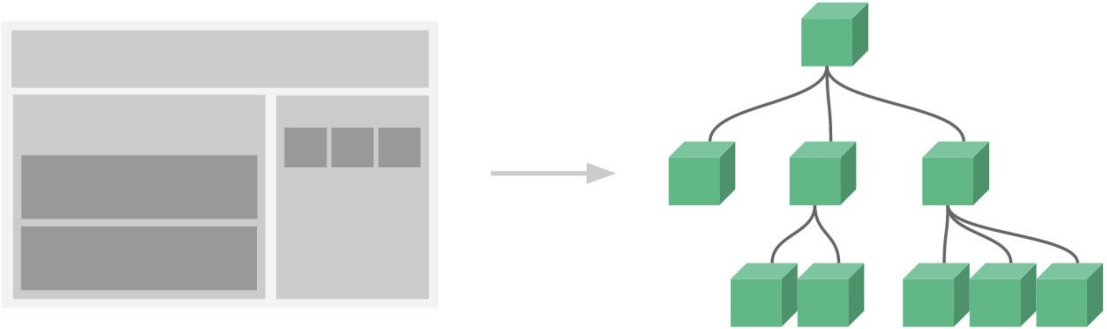
```
<div id="app-5">
  <p>{{ message }}</p>
  <button v-on:click="reverseMessage">Reverse Message</button>
</div>
var app5 = new Vue({
  el: '#app-5',
  data: {
    message: 'Hello Vue.js!'
  },
  methods: {
    reverseMessage: function () {
      this.message = this.message.split('').reverse().join('')
    }
  }
})
```

v-model example

```
<div id="app-6">
  <p>{{ message }}</p>
  <input v-model="message">
</div>
var app6 = new Vue({
  el: '#app-6',
  data: {
    message: 'Hello Vue!'
  }
})
```

Component-based design

- Makes it easier to develop apps starting from their UI
- Like in React, we consider all UIs to be structures in a tree



Vue components

- Vue instances with predefined options (like React components)

```
<ol>
  <!-- Create an instance of the todo-item component -->
  <todo-item></todo-item>
</ol>

// Define a new component called todo-item
Vue.component('todo-item', {
  template: '<li>This is a todo</li>'
})

var app = new Vue(...)
```

Vue components

```
<div id="app-7">
  <ol>
    <todo-item
      v-for="item in groceryList"
      v-bind:todo="item"
      v-bind:key="item.id"
    ></todo-item>
  </ol>
</div>
```

```
Vue.component('todo-item', {
  props: ['todo'],
  template: '<li>{{ todo.text }}</li>'
})

var app7 = new Vue({
  el: '#app-7',
  data: {
    groceryList: [
      { id: 0, text: 'Vegetables' },
      { id: 1, text: 'Cheese' },
      { id: 2, text: 'Whatever else
humans are supposed to eat' }
    ]
  }
})
```

Instance properties and methods

- Different than user-defined properties and methods
- prefixed with $

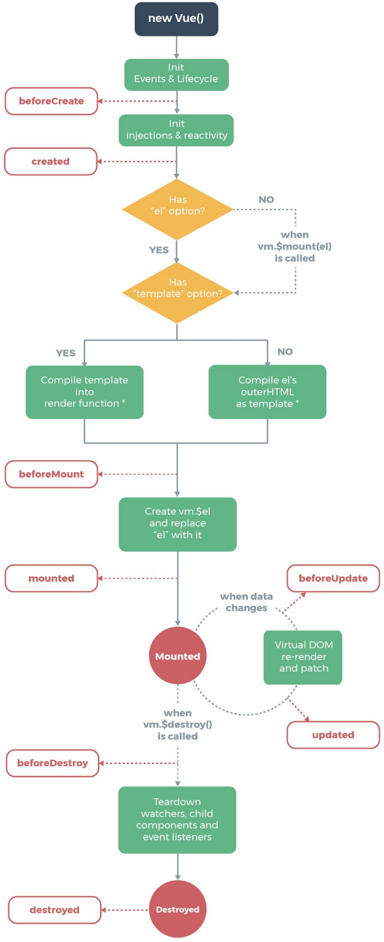
User-defined properties & methods

```
var data = { a: 1 }
var vm = new Vue({
  el:'#example',
  data: data
})

vm.a == data.a // => true

vm.a = 2
data.a // => 2

data.a = 3
vm.a // => 3
```

Instance properties & methods

```
vm.$data === data // => true
vm.$el === document.getElementById('example')
// => true

// $watch is an instance method
vm.$watch('a', function (newValue, oldValue) {
  // This callback will be called when `vm.a`
changes
})
```

Instance lifecycle



* template compilation is performed ahead-of-time if using
a build step, e.g. single-file components

Example

```
<html>
  <head>
    <title>vue el example</title>
    <script
src="/js/vuejs/2.6.10/vue.js"></scr
ipt>
  </head>
  <body>
  <div id="foo">
  </div>
  <script>
new Vue({
     template:'<p>bar</p>',
     created: function(){
         this.$mount('#foo');
     }
  })

  </script>
  </body>
</html>
```

Quizzes

- Create a Vue page that displays the time changing every second.
- Create a Vue page that uses 2 textboxes to convert in real time celsius to fahrenheit and vice versa
- Create a Vue page with a button that once pressed will show “ON” or “OFF” interchangeably

Resources

- https://vuejs.org
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_objects/Function/bind