ОНЛАЙН-ОБРАЗОВАНИЕ

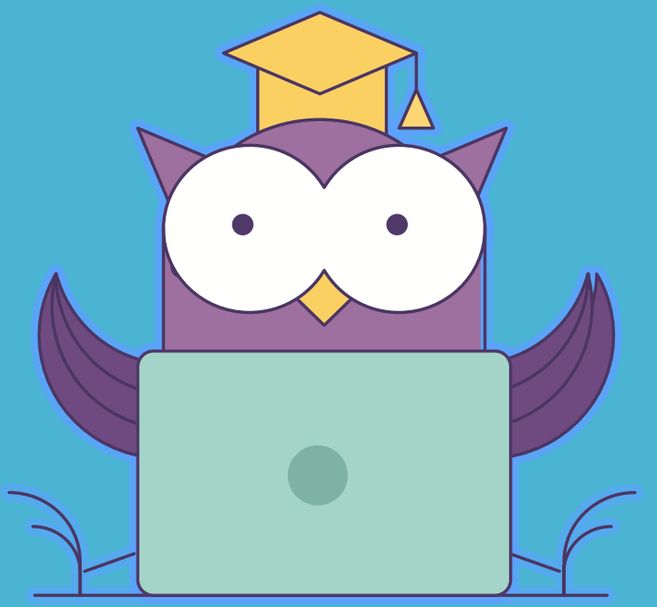
OTUS

Modern JavaScript Frameworks

Все суть компоненты

Александр Коржиков

Как меня слышно и видно?



> Напишите в чат

+ если все хорошо

– если есть проблемы со звуком или с видео

Темы предыдущего занятия

Node Summary

DOM API
Events

DevTools

Темы

- Custom Elements
- Specification
- Standalone Elements
- Built-in Elements
- LifeCycle Hooks

Цели

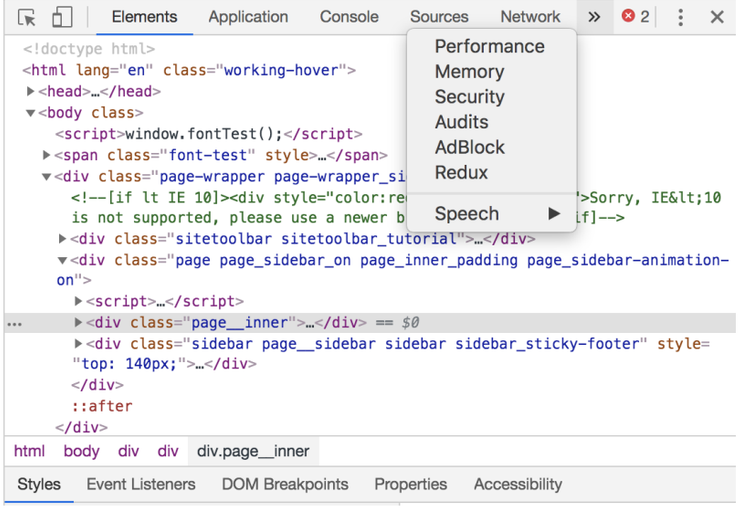
- Понимать и работать с веб спецификацией **Custom Elements**

Docs

- WHATWG Specification
- Polymer Guide
- A short introduction to Web Components
- Google I/O 2014 - Polymer and Web Components change everything you know about Web development

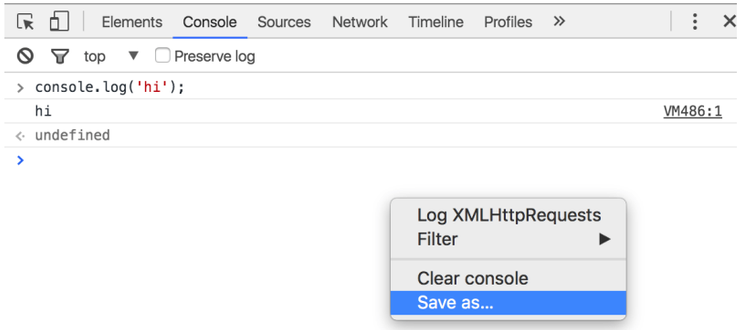
Chrome DevTools - Demo

- Console
- Sources
- Network
- Application
- Device Mode
- ...



Console Tricks

- debugger
- $0
- [$0]
- copy
- table
- Copy > Copy as cURL
- ...?



Задача

Какая функция отработает быстрее?

http://output.jsbin.com/feloni/3/quiet

Custom Elements

"Custom elements provide a way for authors to build their own fully-featured DOM elements"

"A custom element is an element that is custom" 🤔

Все суть компоненты

Например, select, input & form

```
<select>
<option value="1">7</option>
</select>
```

Что если нам нужен multi-select?

```
<multi-select>
  <option value="1">8</option>
  <option value="2">13</option>
</multi-select>
```

Пример 1

```
<script>
class HelloWorldElement extends HTMLElement {
  connectedCallback() {
    this.textContent = "Hello World"
  }
}
customElements.define('hello-world-element', HelloWorldElement);
</script>
```

Как использовать?

Пример 2

```
class FlagIcon extends HTMLElement {
  constructor() {
    super();
    this._countryCode = null;
  }
  attributeChangedCallback(name, oldValue, newValue) {
    this._countryCode = newValue;
  }
  connectedCallback() { /* ... */ }
  get country() {
    return this._countryCode;
  }
  set country(v) {
    this.setAttribute("country", v);
  }
}
```

Декларация

```
customElements.define("flag-icon", FlagIcon)
// [a-z](PCENChar)* '-' (PCENChar)*

// use createElement
const flagIcon = document.createElement("flag-icon")
flagIcon.country = "jp"
document.body.appendChild(flagIcon)

// use new
const flagIcon = new FlagIcon()
flagIcon.country = "jp"
document.body.appendChild(flagIcon)
// use HTML document
```

Самостоятельная работа

Создать **hello-component** элемент, который покажет **alert** на странице

```
class HelloComponent extends HTMLElement { // ?}

// customElements.define ?

<!-- ? -->
```

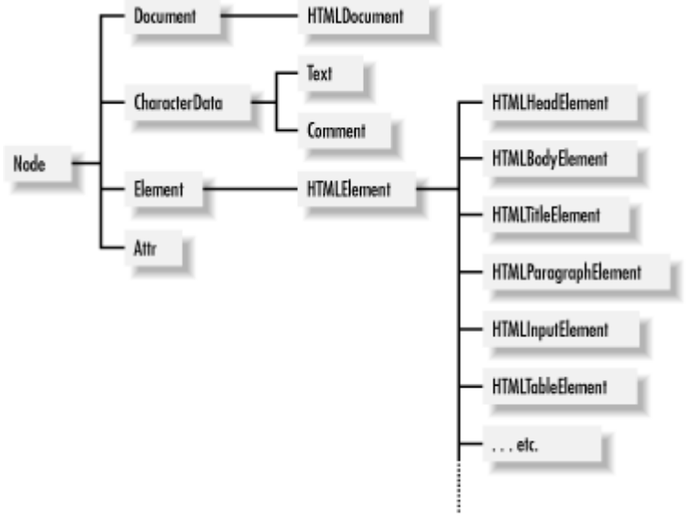
Какое значение у **display hello-component** элемента в **DOM**?

Вопрос

- Сложно ли создать собственный **button**?
- Что для этого нужно сделать?

HTMLElement

```
class HelloComponent extends HTMLElement { }
```



Customized Built-in Elements

- reuse && extend - встроенное поведение
- extends && is - обязательные аттрибуты

```
class PlasticButton extends HTMLButtonElement {
  constructor() {
    super() // ...
  }
}

customElements.define("plastic-button",
  PlasticButton, { extends: "button" }
)

document.createElement("button", {
  is: "plastic-button"
})
```

```
<button is="plastic-button">Click Me!</button>
```

Демо

Создать custom label для активации ссылки

Вопросы

- Напоминает что-нибудь?
- Что бы Вы предложили реализовать для примера **customized built-in elements**?
- Что будет, если декларировать **custom element** после его создания?

(new) Form-Associated Custom Elements

Дополнительный контроль за специфичными событиями форм

```
class MyControl extends HTMLElement {
    static formAssociated = true;
    // name, disabled, form, readonly
}
```

Form-associated custom elements supports additional lifecycle callbacks

- **formAssociatedCallback()** - associated || disassociated the element from a form element
- **formDisabledCallback()** - **<fieldset>**'s disabled state
- **formResetCallback()**

CustomElementRegistry

- **window.customElements** - registry instance
- **define()**
- **get()**
- **whenDefined()**
- **upgrade()**

```
<example-element></example-element>
```

```
const inDocument = document.querySelector('example-element')
const outOfDocument = document.createElement('example-element')

console.assert(inDocument instanceof HTMLElement)
console.assert(outOfDocument instanceof HTMLElement)

class ExampleElement extends HTMLElement {}
customElements.define('example-element', ExampleElement)

console.assert(inDocument instanceof ExampleElement)
console.assert(!(outOfDocument instanceof ExampleElement))

// upgraded
document.body.appendChild(outOfDocument)
console.assert(outOfDocument instanceof ExampleElement)
```

LifeCycle

- **constructor** (0)
- **attributeChangedCallback** (1) <= static **observedAttributes()**
- **connectedCallback** (2) - **DOM**
- **disconnectedCallback** (N) - **DOM**
- **adoptedCallback** (?) => "new document"

Attribute Change

```
class HelloWorldElement extends HTMLElement {
  static get observedAttributes() {
    return ['name']
  }
  attributeChangedCallback(name, oldValue, newValue) {
    this._name = newValue
  }
  connectedCallback() {
    this.name = this.getAttribute('name') || 'World'
  }
  get name() {
    return this._name
  }
  set name(name) {
    this.setAttribute('name', name)
    this.render()
  }
  render() {
    this.textContent = `Hello ${this.name}`
  }
}
```

Самостоятельная работа

Убедиться что все хуки исполнились (с помощью консоли и ...?)

- **connectedCallback**
- **disconnectedCallback**
- **attributeChangedCallback**
- **adoptedCallback**
- **constructor**

- **Custom Elements** can be upgraded
- call **super()**
- work to be deferred to **connectedCallback()**
- constructor for state, default values, event listeners & shadow root

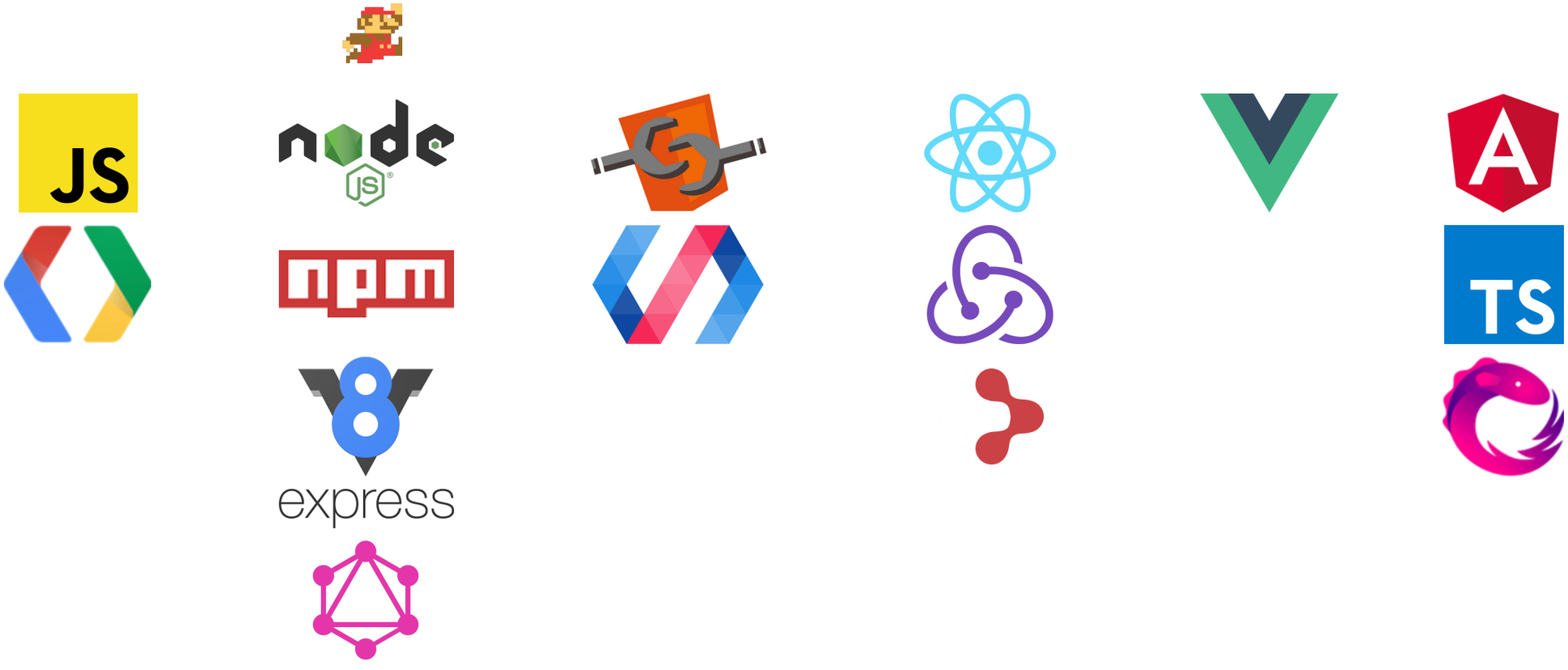
Вопрос

Как передавать данные для дочерних Custom Elements?

На занятии

- Попрактиковали **getters / setters**
- Разобрали веб спецификацию **Custom Elements**

Modern JavaScript Frameworks



Самостоятельная работа

Спасибо за внимание!

Вы верите в Web Components?

Пожалуйста, пройдите опрос в личном кабинете