Google Developer Student Clubs

WebComponents - intro

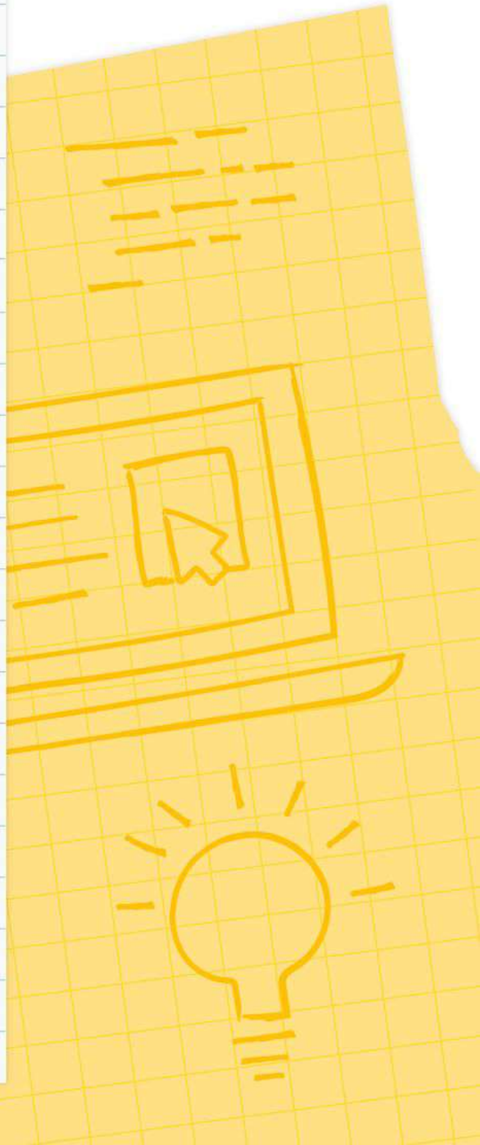
Warsztaty

Krzysztof Kaczyński

Google Developer Student Clubs

Agenda

- Czym są WebComponents?
- Cykle życia custom elementu
- Atrybuty i propertasy
- Tag ***<template>***
- ShadowDOM
- Tag ***<slot>***
- Event model w ShadowDOM
- Stylowanie



Czym są WebComponents?

`Meta specyfikacja stworzona przez 4 specyfikacje`

Custom Elements
Shadow DOM
HTML Template
ES Module

> Podział kodu, reużywalność, większa kontrola ...

Cykle życia

Możemy wyróżnić 5 cykli życia komponentu

- ***Constructor***
- ***connectedCallback***
- ***disconnnectedCallback***
- ***attributeChangedCallback***
- ***adoptedCallback***

Atrybuty i propertasy

Co za różnica ?

Atrybuty to pary kluczy i wartości dostępnych w drzewie DOM kodu HTML

Propertasy to odzwierciedlenie atrybutów w obiektach JavaScript. Zmiana propertasa zazwyczaj wiąże się ze zmianą odpowiedniego atrybutu html

```
div.id = "new-div-id”
<div id="new-div-id"></div>
```

Google Developer Student Clubs

<template>

Nowy tag HTML

- Umożliwia tworzenie niewidocznego z zewnątrz wrappera szablonu html dla komponentu
- Szablon HTML wewnątrz ***<template>*** jest parsowany raz i może być wielokrotnie użyty

Przykład 1:

```
const template = '<h1>SUPER</h1>';

class CustomEL1 extends HTMLElement {

 constructor() {
   super();
   this.attachShadow({mode: 'open'});
   this.shadowRoot.innerHTML = template;
 }

}
```

Przykład 2:

```
const template = document.createElement('template');
template.innerHTML = '<h1>SUPER</h1>';

class CustomEL2 extends HTMLElement {

 constructor() {
   super();
   this.attachShadow({mode: 'open'});
   this.shadowRoot.append(template.content.cloneNode(true));
 }

}
```

Shadow DOM

- Izolowany DOM
- Scoped CSS
- Prostsze selektory CSS
- Większa kontrola nad kodem

```
const template = `
<div class="box">
 <span class="info"></span>
</div>
`;

export class InfoBox extends KKWebComponent {
 static TAG = `kk-info-box`;
 static observedAttributes = ['kk-content'];

 contentWrapper = this.shadowRoot.querySelector('.info');

 constructor(infoContent) {
   super(template, style);
   if(infoContent) {
     this.content = infoContent;
   }
 }

 get content() {
   return this.contentWrapper.textContent;
 }

 set content(value) {
   this.contentWrapper.textContent = value;
 }

 attributeChangedCallback(name, oldValue, newValue) {
   if (oldValue === newValue) {
     return void 0;
   }
   switch (name) {
     case 'kk-content':
       this.content = newValue;
       break;
     default:
       throw new Error(`Attribute ${name} doesn't exist in ${InfoBox.name} component`);
   }
 }

}

customElements.define(InfoBox.TAG, InfoBox);
```

Google Developer Student Clubs

Tag <slot>

- Wstrzykiwanie kodu HTML przez osobę wykorzystującą WebComponent
- Możliwość przygotowania bardzo generycznej templatki z przygotowanymi miejscami, które mogą zostać podmienione przez użytkownika

```
<p><slot name="slot-id">My default text</slot></p>
```

```
<my-paragraph>
 <span slot="slot-id">Let's have some different text!</span>
</my-paragraph>
```

Event model w ShadowDOM

W ShadowDOM eventy są nadawane w taki sposób, żeby zawsze wyglądało to tak, że to nasz niestandardowy komponent wyemitował event.

Niektóre eventy nie przechodzą przez barierę ShadowDOM.

Jeżeli emitujemy **CustomEvent**. Wtedy należy pamiętać, aby w dodatkowych opcjach, przekazać flagę **composed** ustawioną na *true*. W przeciwnym razie **CustomEvent** nie przeniknie przez barierę ShadowDOM.

Stylowanie

- **:host**
- **:host-context(<css-selector>)**
- **:slotted(<css-selector>)**
- **contain: content**
- **<link rel="stylesheet" ... />**
- Display inline jest domyślnym trybem
- Css variables

```
:root {
 --color-primary: #272727;
 --color-primary-dark: #202124;
 --color-primary-light: #333;
 --color-accent-1: #3fba84;
 --color-accent-2: #263238;
 --color-text: #d6d6d6;
 --color-text-on-accent-1: #000;
}

:host {
 display: block;
 contain: content;
}

:host(:hover) {
 opacity: 1;
}

:host(.blue) {
 color: blue;
}

:host-context(.darktheme) {
 color: white;
 background: black;
}

::slotted(h2) {
 margin: 0;
 font-weight: 300;
 color: red;
}

::slotted(.title) {
 color: orange;
}
```

DZIĘKUJĘ ZA UWAGĘ