



*VSR://edu/2020/evs/*

# 07 – Web Components

*//// Design of Distributed Systems //////////////////////////////////////*

**M.Sc Mahda Noura**

*VSR.Informatik.TU-Chemnitz.de*



# Web Components

# Web components promote:

- Encapsulation
- Reuse

# Web components specifications:

- HTML Imports
- HTML Templates
- Custom Elements
- Shadow DOM



# Recipe

## Define Class in JS

- Regular JS Class
- Extends HTMLElement
- Always call super() in constructor
- Use pre-defined callbacks if needed (connectedCallback, attributeChangedCallback)

```
class ShoppingCart extends HTMLElement {
```

# Define the custom element

- Call `window.customElements.define()` to link tag name of custom element to class name of implementation

```
window.customElements.define('shopping-cart', ShoppingCart);
```

# Define HTML Template

- Use HTML template tag to define the HTML structure of the Web Component

```
<template id="shopping-template">
```



# Instantiate Shadow DOM

- In constructor, select the template
- Create a Shadow Dom
- Copy the template nodes to the Shadow Root

```
let tmpl = document.querySelector('#shopping-template');  
let shadowRoot = this.attachShadow({mode: 'open'});  
shadowRoot.appendChild(tmpl.content.cloneNode(true));
```

# Use Custom Element in HTML

- Name MUST contain dash
- Cannot be self-closed
- Cannot register the same tag more than once

```
<shopping-cart></shopping-cart>
```

# Use HTML Attributes for Parameter Passing

- Add attributes to Custom Element

```
<shopping-cart endpoint="http://localhost:3000/cart" ></shopping-cart>
```

- Read/Write attributes from JS Class

```
get endpointUrl() {  
    if (!this.hasAttribute('endpoint'))  
        alert('Endpoint URL not set');  
    return this.getAttribute('endpoint');  
}
```

# Access DOM Nodes within Custom Elements

- *this* points to the Web Component
- *Use `this.shadowRoot.getElementById()`, `.querySelector()` etc. to access Shadow DOM Nodes*

```
let submit_button = shadowRoot.querySelector('button');
submit_button.addEventListener('click', e => {
  this.addNewEntry();
  e.preventDefault();
});
```



# Tasks

# Task 1: Create a custom component for adding and removing items to/from the shopping cart



VSR

Your feedback on today's session:



**Questions?**

[mytuc.org/tgxs](https://mytuc.org/tgxs)

**mahda.noura@informatik.tu-chemnitz.de**

*VSR.Informatik.TU-Chemnitz.de*