

Introduction to DOM and Events

Create / Delete DOM Elements
Handle Browser Events



SoftUni Team
Technical Trainers
Software University
<http://softuni.bg>

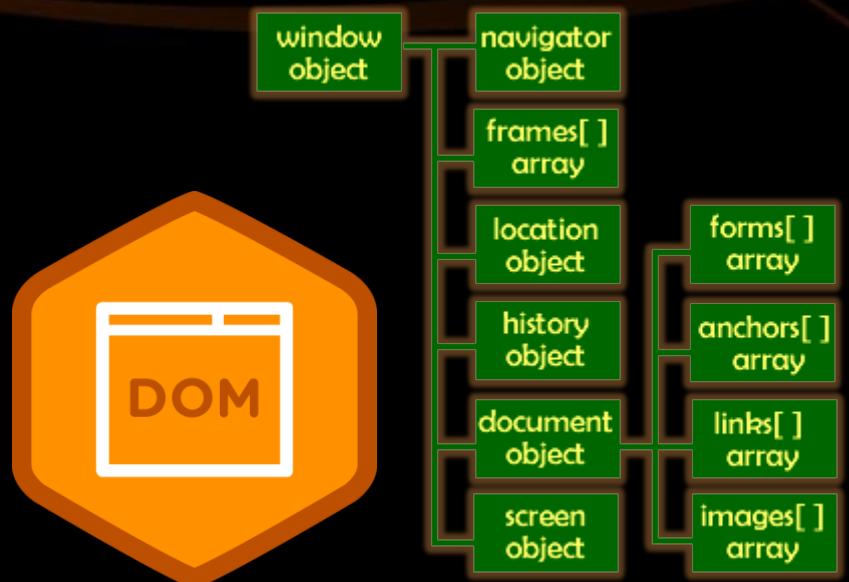


Table of Contents

1. What is DOM?
2. Manipulating the DOM
 - Create Elements
 - Delete Elements
 - Element Attributes
3. Event Handling
 - Attach / Detach Events

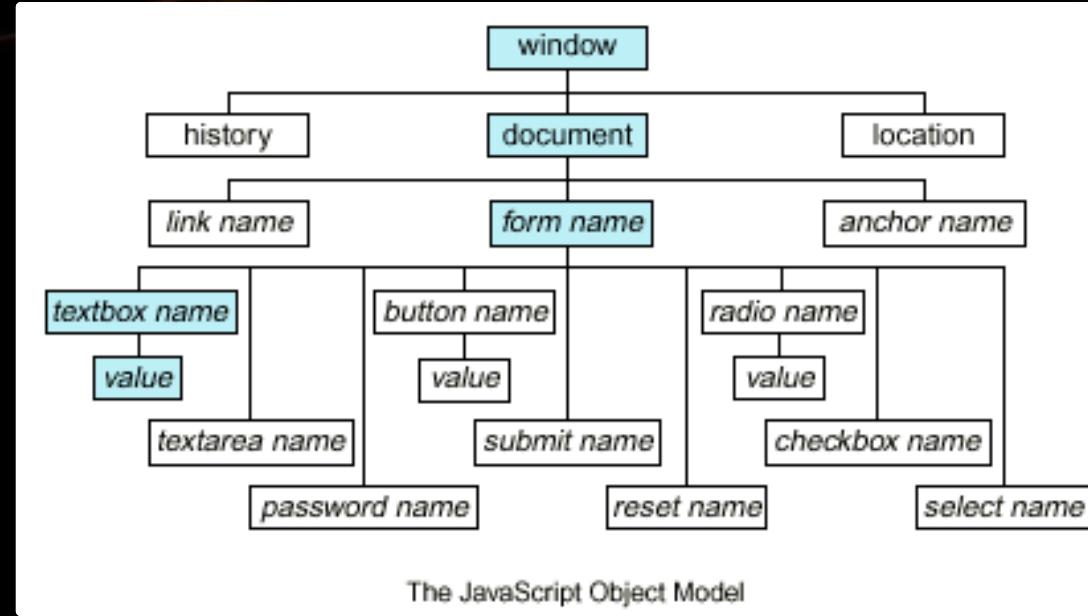


Have a Question?



sli.do

#front-end



Document Object Model (DOM)

Traverse the DOM Tree

What is DOM?

- What is Document Object Model (DOM)?
 - HTML documents in the browser are stored as "DOM tree"
 - Consists of elements with child elements
 - Elements have properties (attribute + value) and events
- The DOM API allows search / modify the DOM tree

```
let menu = document.getElementById('menu');
menu.style.display = 'none';
menu.appendChild(document.createElement('hr'));
```

Selecting HTML Elements from DOM

- Select a single element → returns **HTML Element**

```
let header = document.getElementById('header');
let nav = document.querySelector('#main-nav');
let root = document.documentElement;
```

- Select a collection of elements → returns a **collection**

```
let inputs = document.getElementsByTagName('li');
let towns = document.getElementsByName('towns[]');
let header = document.querySelectorAll('#nav li');
let allLinks = document.links;
```

Accessing Element Text

- The contents of HTML elements are stored in text nodes
 - To access the contents of an element:

```
let element = document.getElementById('main');
let text = element.textContent; //This is JavaScript!
element.textContent = "Welcome to the DOM";
```

```
<html>
  <head></head>
  <body>
    <div id="main">This is JavaScript!</div>
  </body>
</html>
```



```
<html>
  <head></head>
  <body>
    <div id="main">Welcome to the DOM</div>
  </body>
</html>
```

- If the element has children, returns all text concatenated

Accessing Element Values

- The values of input elements are string properties on them:

```
<html>
  <head></head>
  <body>
    <div id="main">
      <p>Welcome to the DOM</p>
      <input id="num1" type="text">
    </div>
  </body>
</html>
```

```
type: "text"
useMap: ""
validationMessage: ""
▶ validity: ValidityState
  value: "56"
  valueAsNumber: NaN
▶ webkitEntries: Array[0]
  webkitdirectory: false
  width: 0
```

```
let element = document.getElementById('num1');
let num = Number(element.value);
element.value = 56;
```

DOM API: Element Properties

```
<div id="first" class="big">First <b>DIV</b></div>
```

```
let div = document.getElementById('first');
console.log(div.id); // first
console.log(div.tagName); // DIV
console.log(div.className); // big
console.log(div.textContent); // First DIV
console.log(div.innerHTML); // First <b>DIV</b>
console.log(div.outerHTML); // <div id="first"
                           // class="big">First <b>DIV</b></div>
```

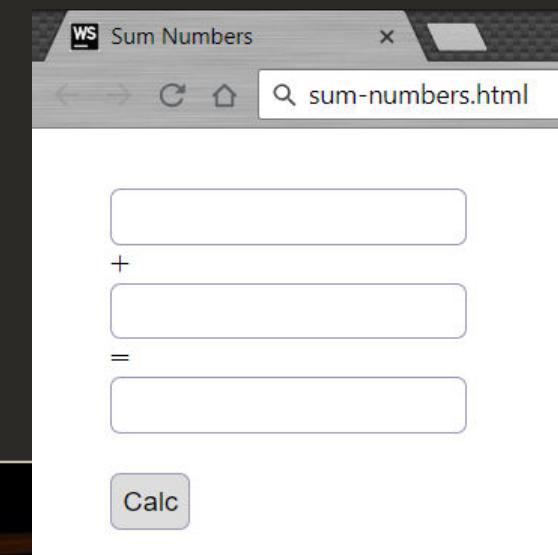
First DIV

Problem: Sum Numbers

- Write a JS function to sum two numbers (fill the missing code)

```
<input type="text" id="num1" /> +
<input type="text" id="num2" /> =
<input type="text" id="sum" readonly="readonly" />
<input type="button" value="Calc" onclick="calc()" />

<script>
    function calc() {
        // TODO: sum = num1 + num2
    }
</script>
```



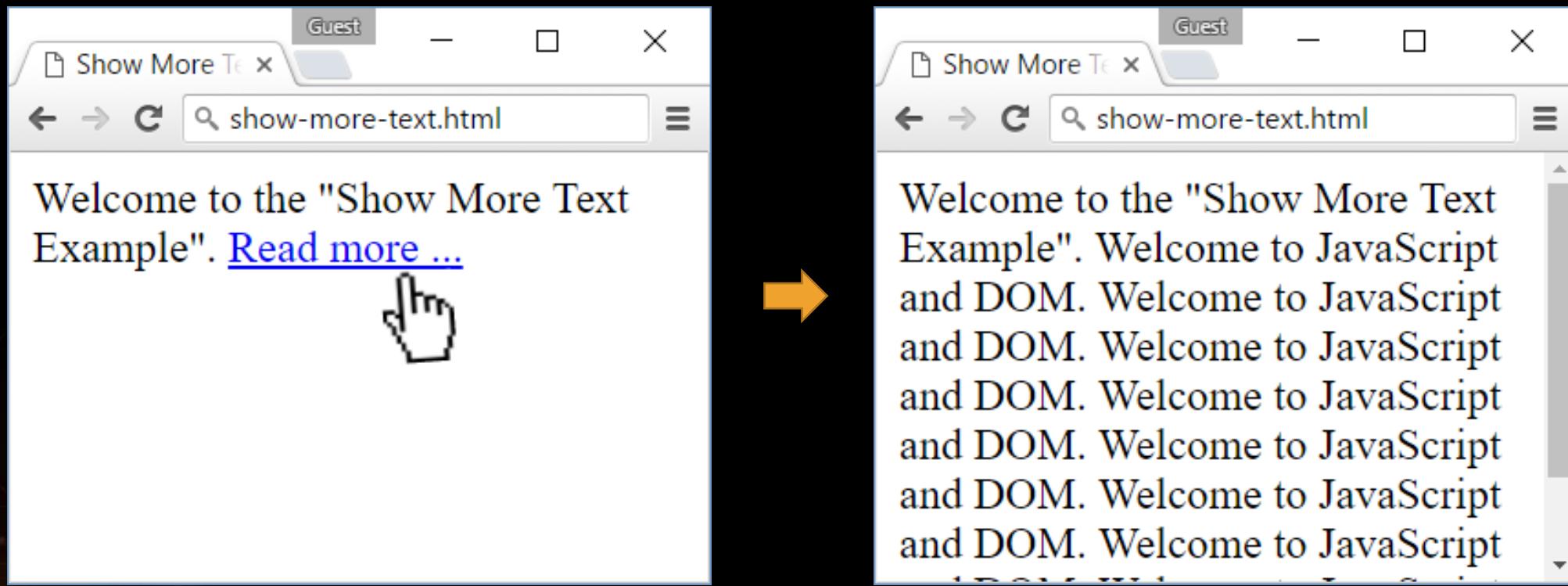
Solution: Sum Numbers

```
function calc() {  
    let num1 = document.getElementById('num1').value;  
    let num2 = document.getElementById('num2').value;  
    let sum = Number(num1) + Number(num2);  
    document.getElementById('sum').value = sum;  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/1096>

Problem: Show More Text

- A HTML page holds a short text + link "*Read more ...*"
 - Clicking on the link shows more text and hides the link



Problem: Show More Text – HTML

Welcome to the "Show More Text Example".

```
<a href="#" id="more" onclick="showText()">Read more ...</a>  
  
<span id="text" style="display:none">Welcome to ...</span>  
  
<script>  
    function showText() {  
        // TODO  
    }  
</script>
```



- See the DOM tree here:
[http://software.hixie.ch/
utilities/js/live-dom-
viewer/?saved=4275](http://software.hixie.ch/utilities/js/live-dom-viewer/?saved=4275)

Solution: Show More Text

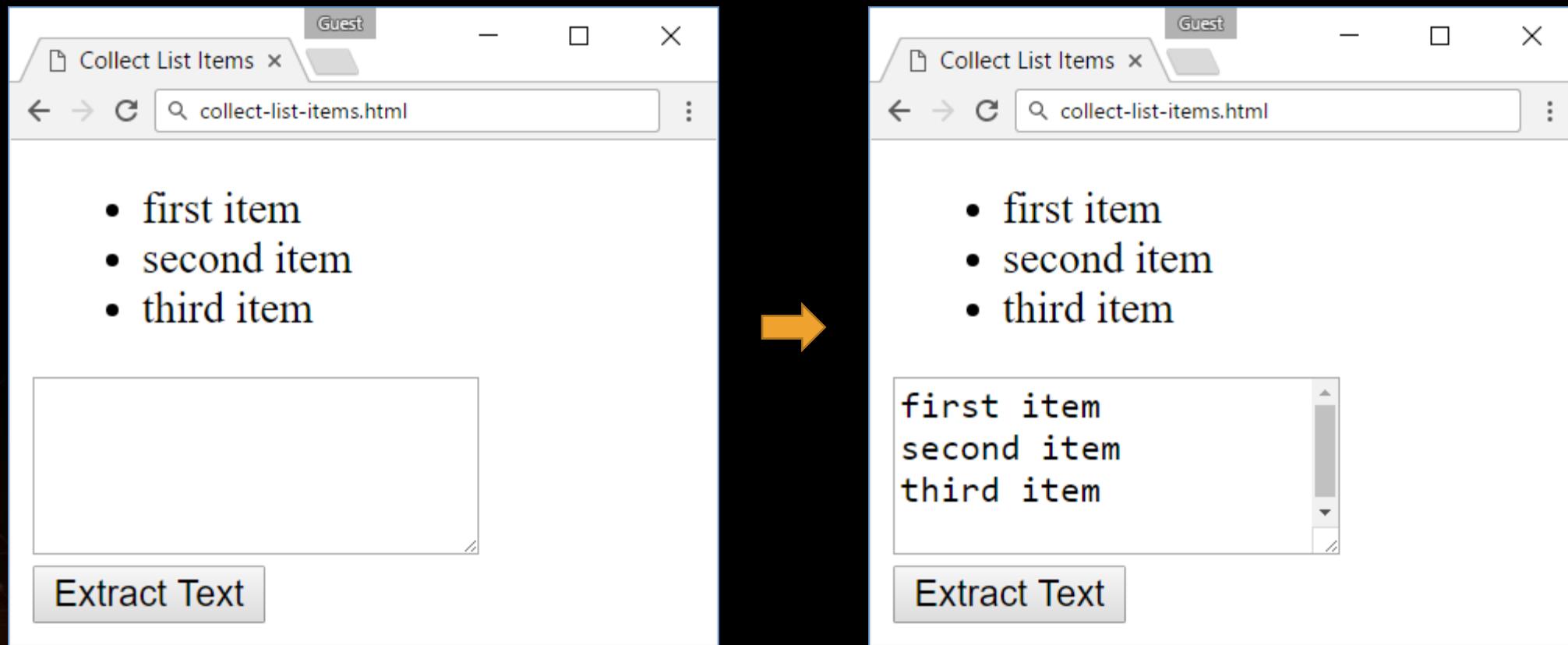
```
Welcome to the "Show More Text Example". <a href="#"  
id="more" onclick="showText()">Read more ...</a>  
<span id="text" style="display:none">Welcome to ...</span>  
<script>  
    function showText() {  
        document.getElementById('text')  
            .style.display = 'inline';  
        document.getElementById('more')  
            .style.display = 'none';  
    }  
</script>
```



Check your solution here: <https://judge.softuni.bg/Contests/1096>

Problem: Collect List Items

- Collect the list items from given HTML list and append their text to given text area



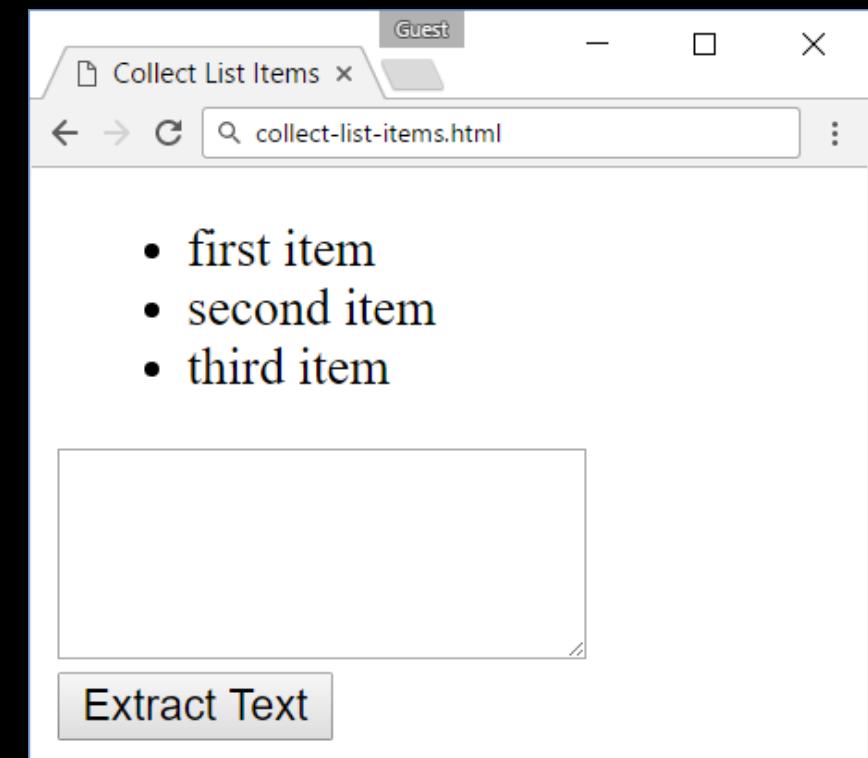
Problem: Collect List Items – HTML

```
<ul id="items">
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ul>

<textarea id="result">
</textarea>

<br>

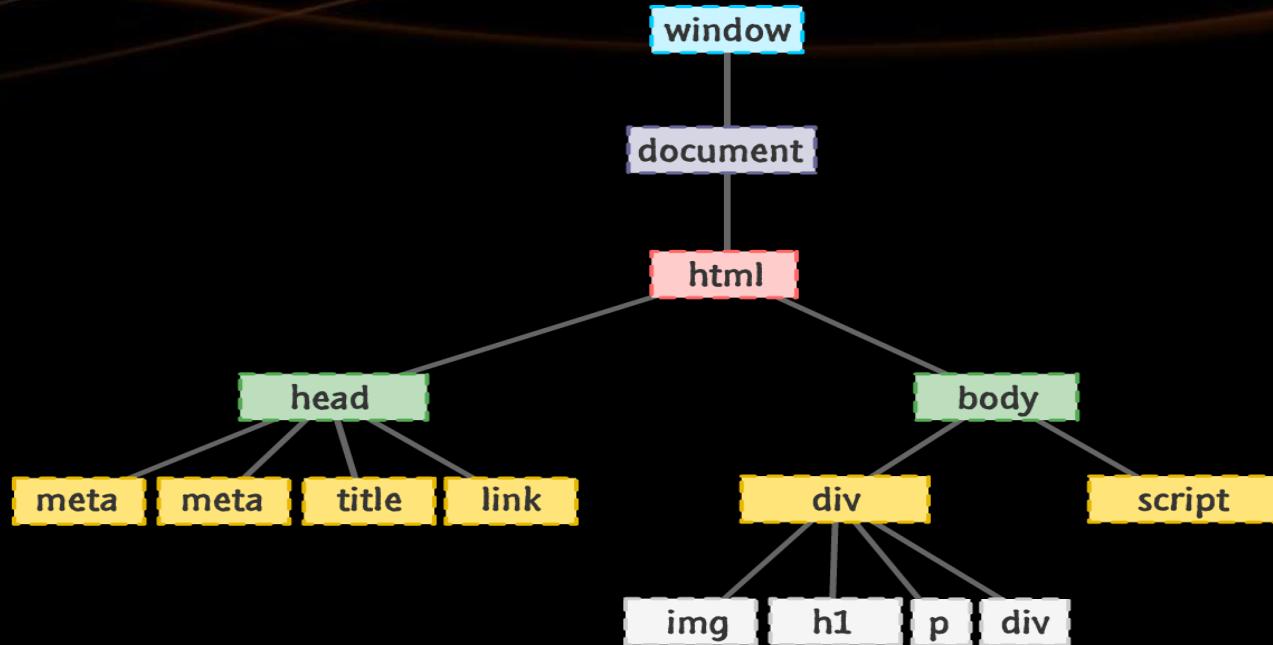
<button onclick="extractText()">
Extract Text</button>
```



Solution: Collect List Items

```
function extractText() {  
    let itemNodes =  
        document.querySelectorAll("ul#items li");  
    let textarea =  
        document.querySelector("#result");  
    for (let node of itemNodes) {  
        textarea.value += node.textContent + "\n";  
    }  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/1096>



DOM Manipulation

Modify the DOM Tree

Creating New DOM Elements

- HTML elements are created with `document.createElement`
- Variables holding HTML elements are `live`:
 - If you `modify` the contents of the variable, the DOM is `updated`
 - If you `insert` it somewhere in the DOM, the original is `moved`
- Text added to `textContent` will be `escaped`
- Text added to `innerHTML` will be `parsed` and turned into actual HTML elements

Creating New DOM Elements: Examples

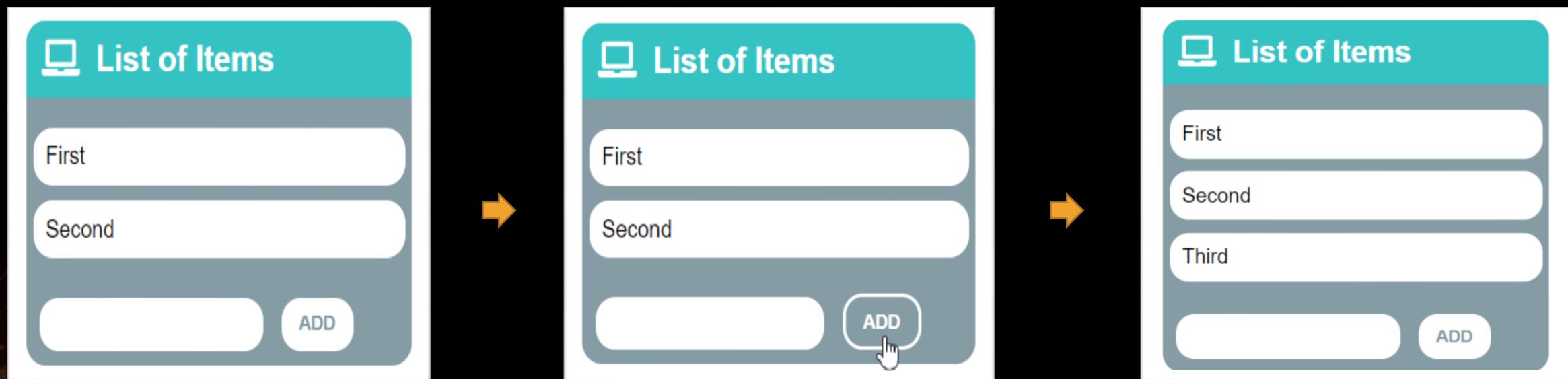
```
let list = document.createElement("ul");
let liPeter = document.createElement("li");
liPeter.textContent = "Peter";
list.appendChild(liPeter);
```

```
let liMaria = document.createElement("li");
liMaria.innerHTML = "<b>Maria</b>";
list.appendChild(liMaria);
document.body.appendChild(list);
```

```
▼<ul>
  <li>Peter</li>
  ▼<li>
    <b>Maria</b>
  </li>
</ul>
```

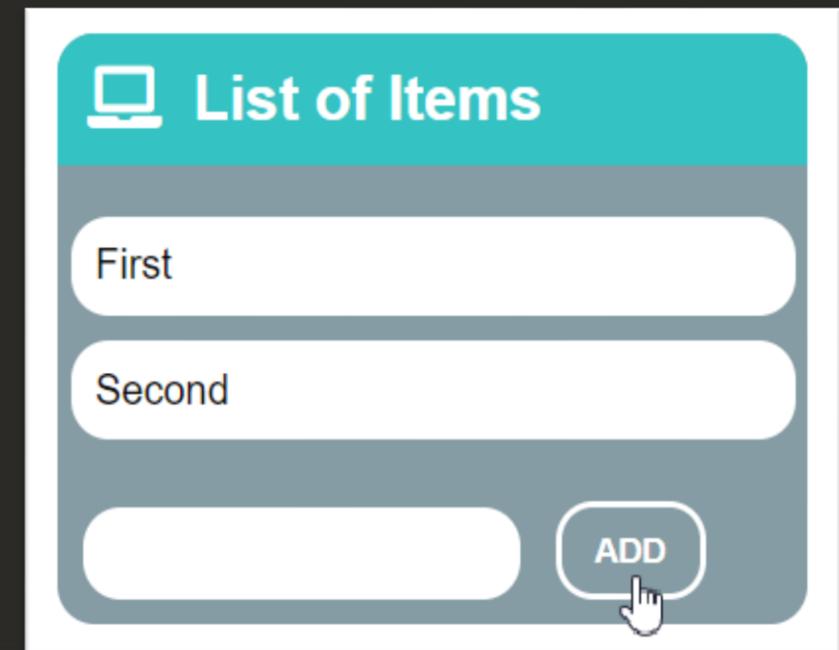
Problem: List of Items

- Create a HTML page holding a list of items + text box + button for adding more items to the list
- Write a JS function to append the specified text to the list



Problem: List of Items – HTML

```
<h1>List of Items</h1>
<ul id="items"><li>First</li><li>Second</li></ul>
<input type="text" id="newItemText" />
<input type="button" value="Add"
      onclick="addItem()">
<script>
  function addItem() {
    // TODO: add new item to the list
  }
</script>
```



Solution: List of Items

```
function addItem() {  
    let text =  
        document.getElementById('newItemText').value;  
    let li = document.createElement("li");  
    li.appendChild(document.createTextNode(text));  
    document.getElementById("items").appendChild(li);  
    document.getElementById('newItemText').value = '';  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/1096>

Deleting DOM Elements

```
<ul id="items">
  <li class="red">Red</li>
  <li class="blue">Blue</li>
</ul>
```

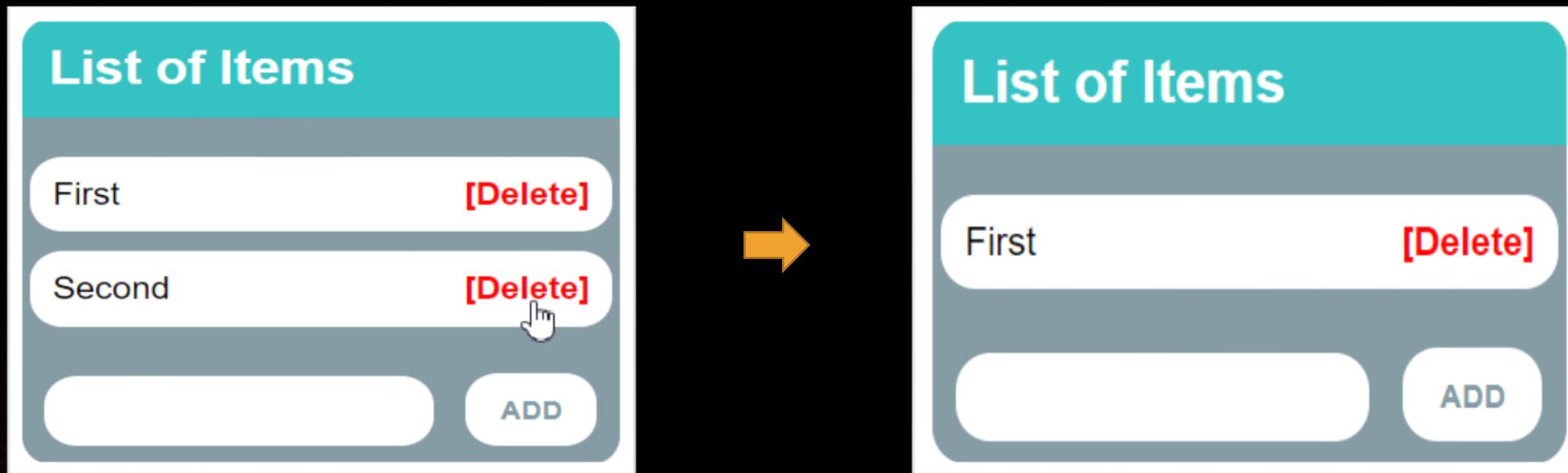
```
▼<body>
  ▼<ul id="items">
    <li class="red">Red</li>
    <li class="blue">Blue</li>
  </ul>
</body>
```

```
let redElements =
  document.querySelectorAll("#items li.red");
redElements.forEach(li => {
  li.parentNode.removeChild(li);
});
```

```
▼<body>
  ▼<ul id="items">
    <li class="blue">Blue</li>
  </ul>
</body>
```

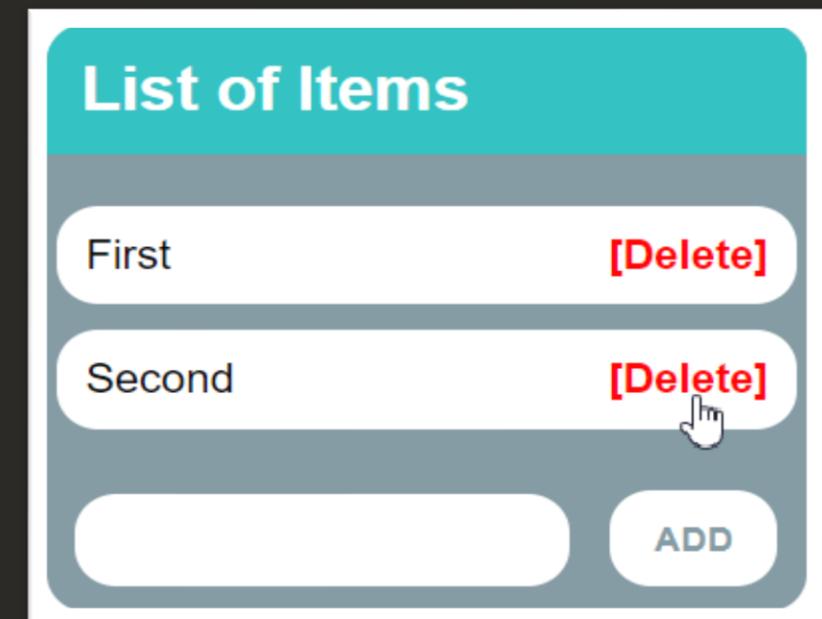
Problem: Add / Delete Items

- Extend the previous problem
 - Implement **[Delete]** action as link after each list item



Problem: Add / Delete Items – HTML

```
<h1>List of Items</h1>
<ul id="items"></ul>
<input type="text" id="newText" />
<input type="button" value="Add"
      onclick="addItem()">
<script>
  function addItem() { ...
    function deleteItem() { ... }
  }
</script>
```

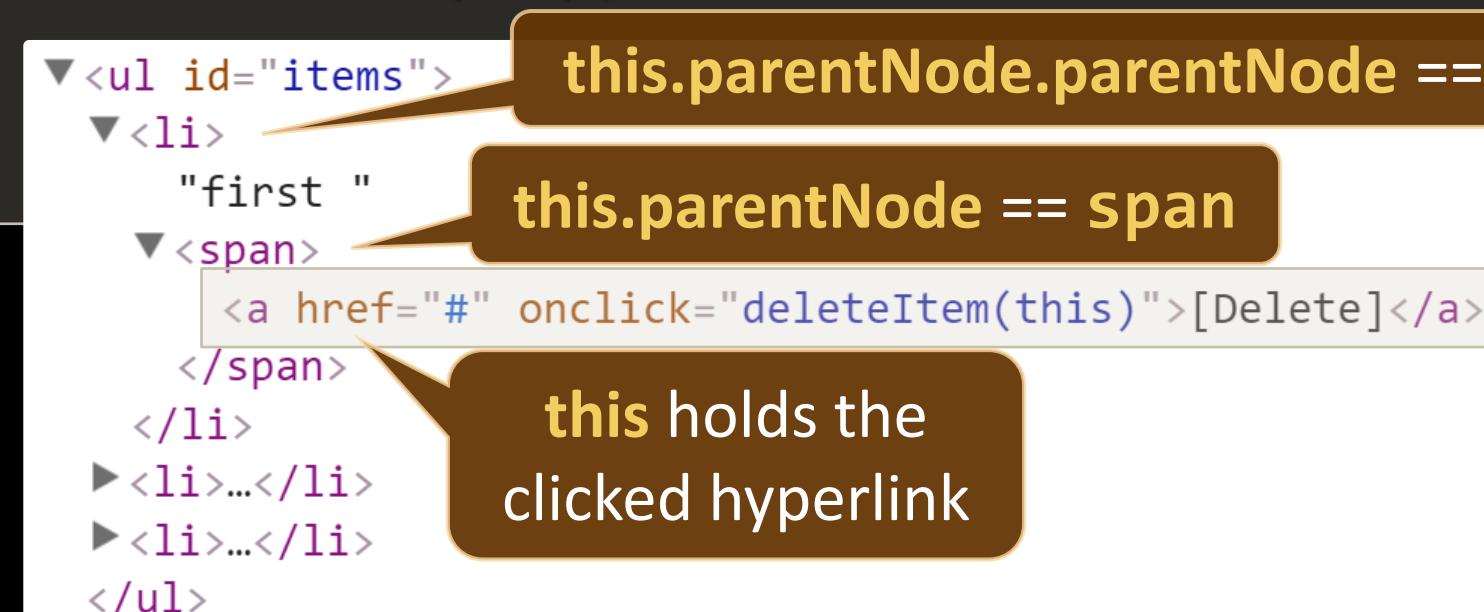


Solution: Add / Delete Items

```
function addItem() {  
    let text = document.getElementById('newText').value;  
    let li = document.createElement("li");  
    li.appendChild(document.createTextNode(text + " "));  
    let span = document.createElement('span');  
    span.innerHTML = "<a href='#'>[Delete]</a>";  
    span.firstChild.addEventListener('click', deleteItem);  
    li.appendChild(span);  
    document.getElementById("items").appendChild(li);  
    document.getElementById('newText').value = '';
```

Solution: Add / Delete Items (2)

```
function deleteItem() {  
    let li = this.parentNode.parentNode;  
    let ul = li.parentNode;  
    ul.removeChild(li);  
}  
}
```



The diagram illustrates the DOM structure of a list item. It shows a tree of nodes:

- <ul id="items">
- "first "
-
 - [Delete]

Three callout boxes point to specific parts of the code and the DOM structure:

- A box points to `this.parentNode.parentNode == li` with the text: `this.parentNode.parentNode == li`.
- A box points to `this.parentNode == span` with the text: `this.parentNode == span`.
- A box points to the `span` element in the DOM structure with the text: `this holds the clicked hyperlink`.

Check your solution here: <https://judge.softuni.bg/Contests/1096>



Practice: DOM and Events

Live Exercises in Class (Lab)

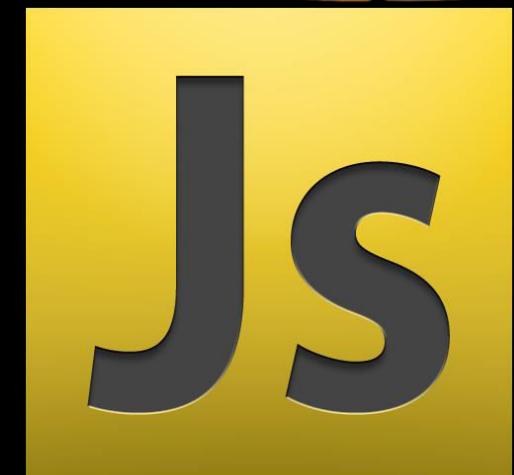
Summary

- Browsers keep elements as DOM tree
- Finding / modifying DOM elements:

```
let menu = document.getElementById('menu');
menu.style.display = 'none';
menu.appendChild(
  document.createElement('hr'));
```

- Handling events:

```
let menu = document.getElementById('menu');
menu.onclick = function(event) { ... }
```



JS for Front-end – Intro to DOM and Events



Questions?



SoftUni Diamond Partners



INDEAVR
Serving the high achievers

 **INFRASTURICS®**



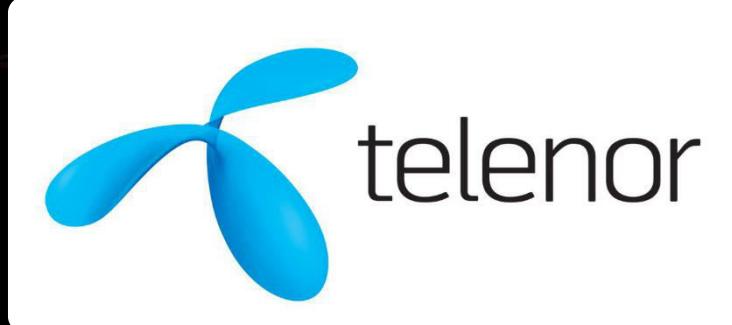
SoftwareGroup
doing it right



NETPEAK
SEO and PPC for Business

**SUPER
HOSTING**
®.BG

SoftUni Diamond Partners



LIEBHERR



License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

