

Document Object Model



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

DOM and BOM, Traverse DOM Tree



Table of Contents

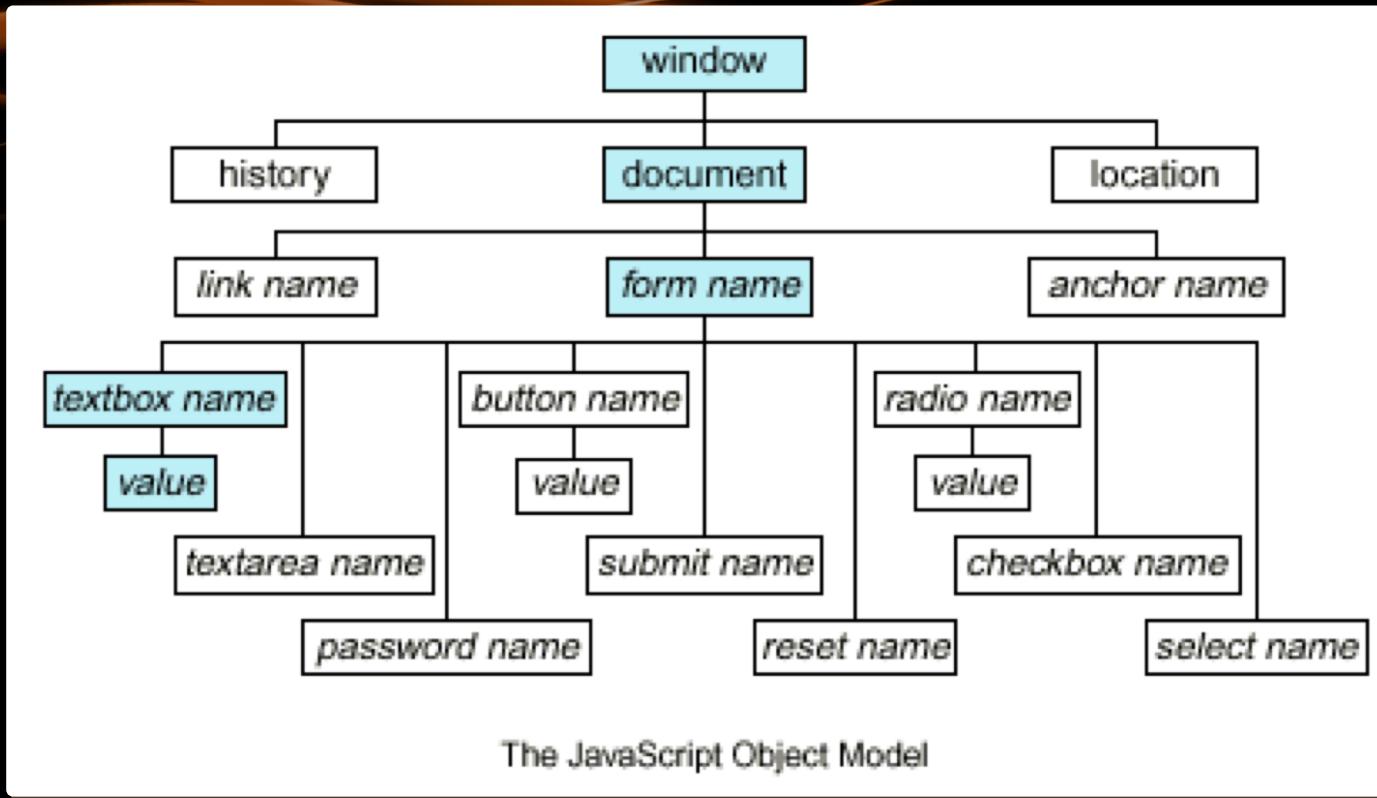
1. What is DOM?
2. Using the DOM API
 - Select Elements
3. What is BOM?
 - Using Timers



Have a Question?



sli.do
#JSCORE



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 **HTMLElement**

```
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 HTML

- To access raw HTML:

```
element.innerHTML = "<p>Welcome to the DOM</p>";
```

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



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

- This will be parsed – beware of XSS attacks!
- Changing **textContent** or **innerHTML** removes all child nodes

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" 56
  valueAsNumber: NaN
▶ webkitEntries: Array[0]
  webkitdirectory: false
  width: 0
```

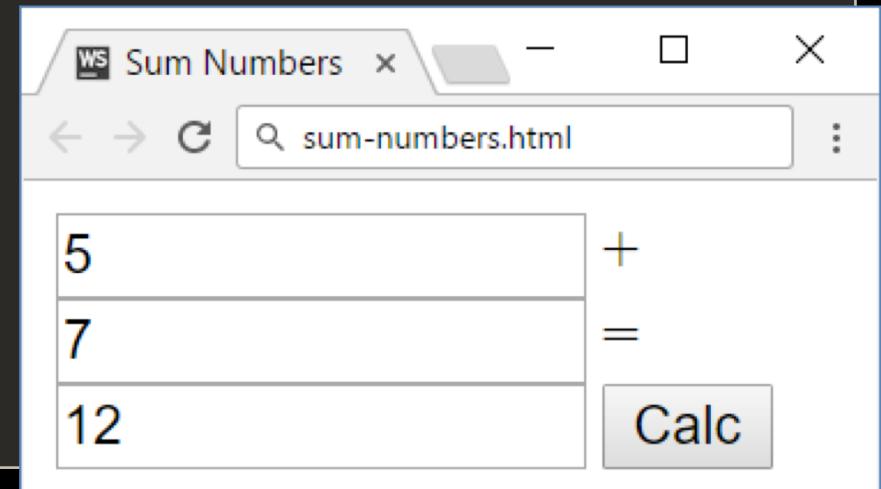
```
let element = document.getElementById('num1');
let num = Number(element.value);
element.value = 56;
```

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/328>

DOM API: Element Properties

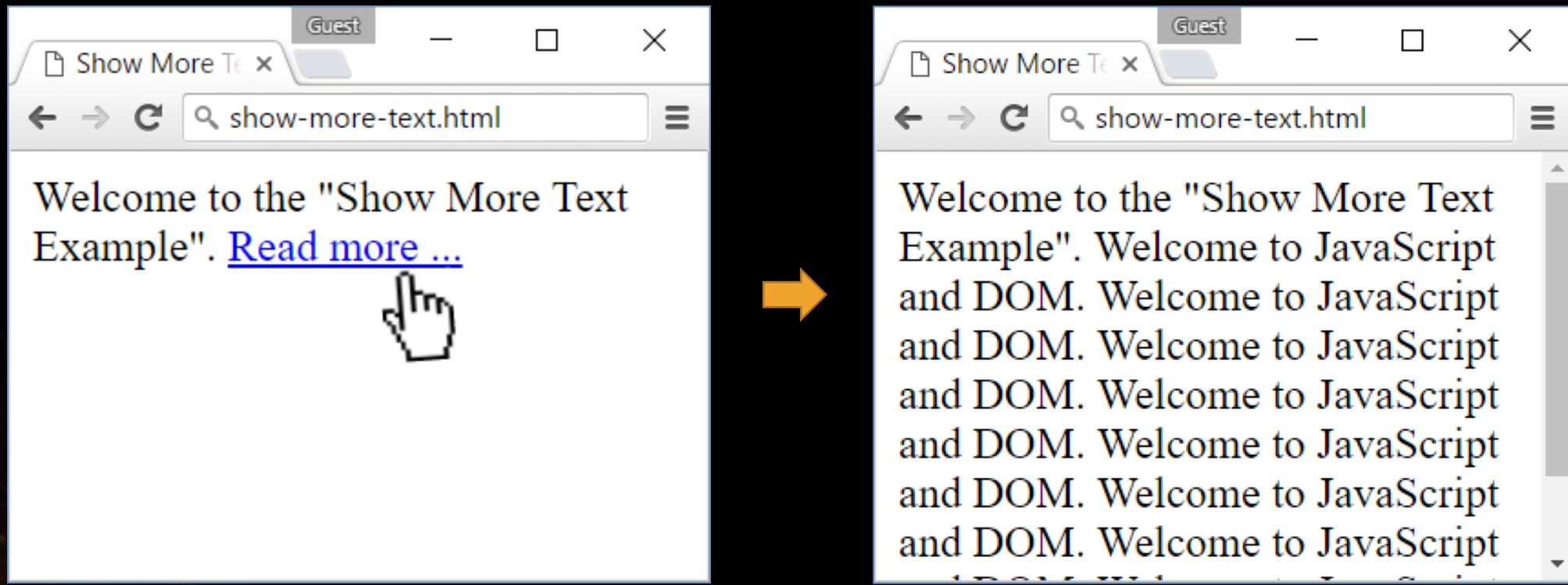
```
<div id="first" class="big">First <b>DIV</b></div>
```

```
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: 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>
```



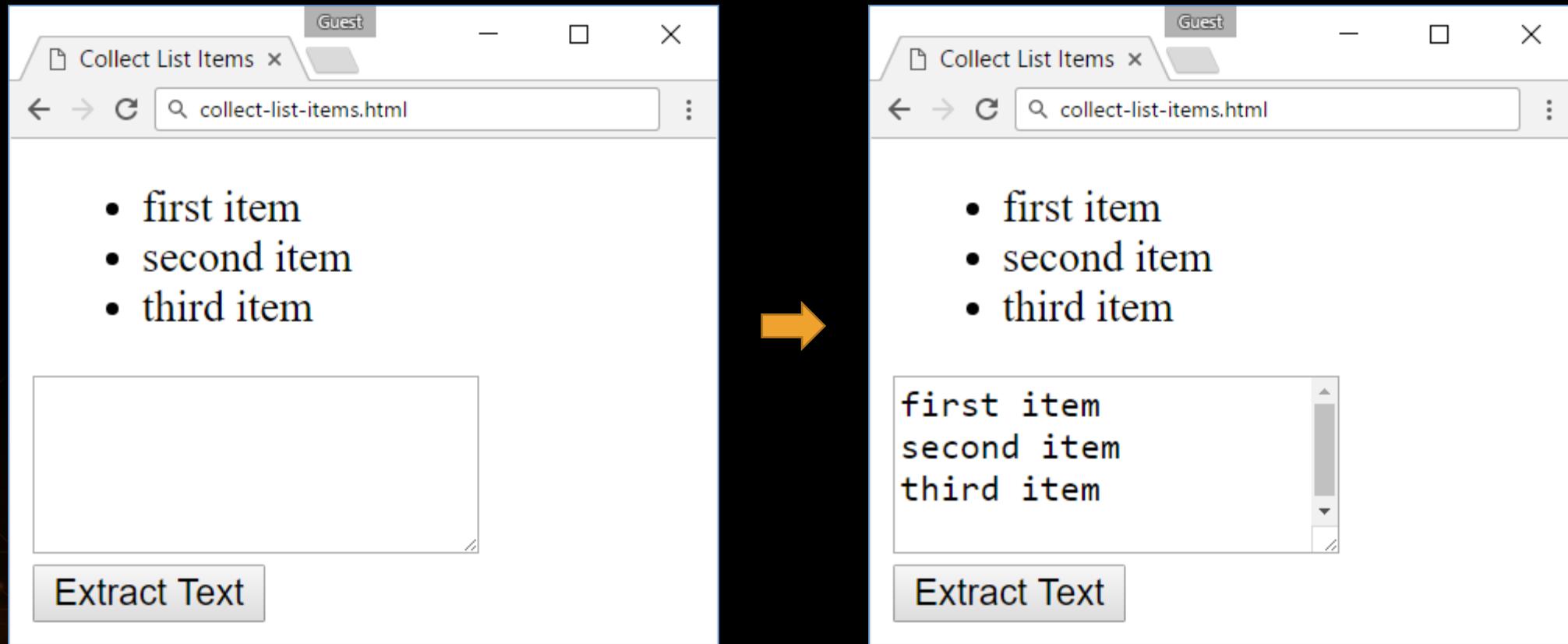
Check your solution here: <https://judge.softuni.bg/Contests/328>

CSS Selectors

- CSS selectors are strings that follow CSS syntax for matching
 - Works with **querySelector** and **querySelectorAll**
 - Base for **jQuery** operations (helper library)
- They allow very fast and powerful element matching, e.g.:
 - "#main" → returns the element with ID "main"
 - "#content div" → selects all **<div>**s inside **#content**
 - ".note, .alert" → all elements with class "note" or "alert"
 - "input[name='login']" → **<input>** with name "login"

Problem: Collect List Items

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



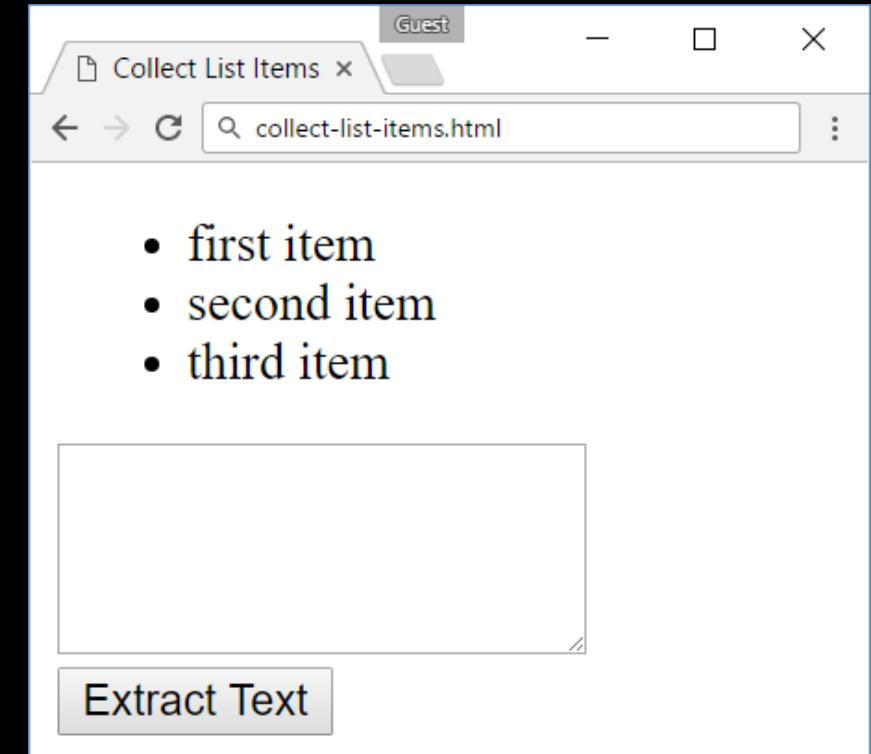
Problem: Collect List Items – HTML

```
<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";  
    }  
}
```

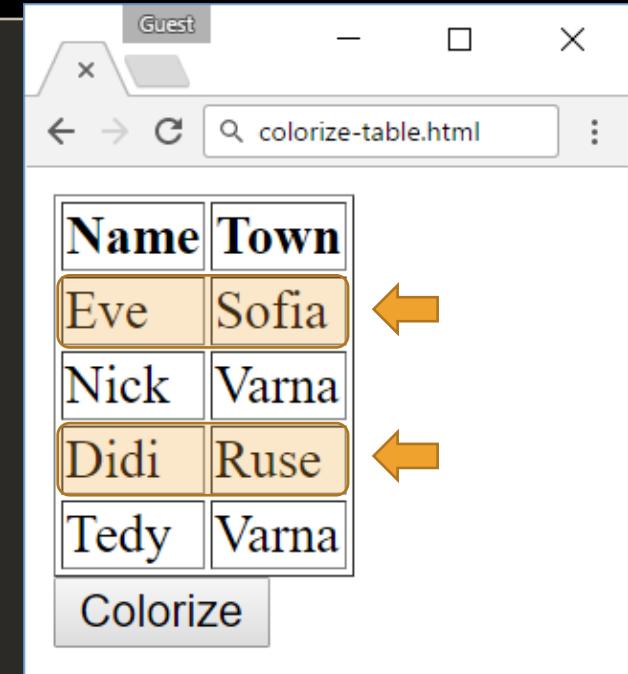
Check your solution here: <https://judge.softuni.bg/Contests/328>

Problem: Colorize Table Rows

- A HTML page holds a table with rows
 - On button click, colorize in color "teal" all even rows

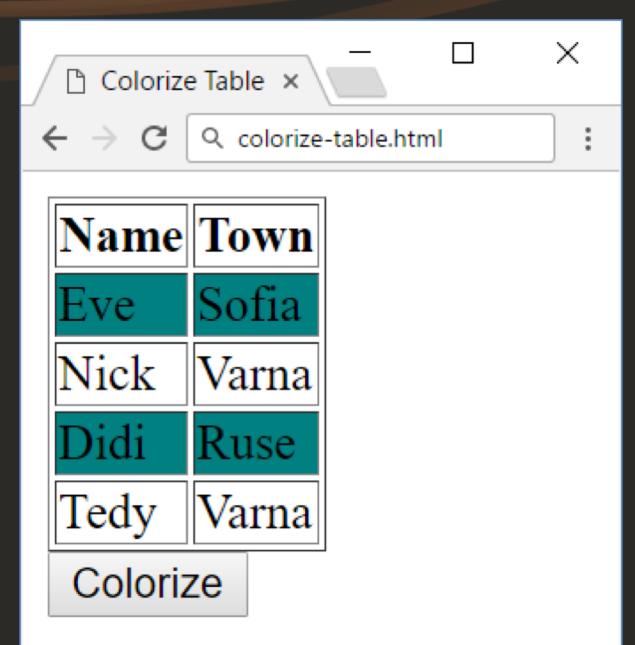
```

<table border="1">
  <tr><th>Name</th><th>Town</th></tr>
  <tr><td>Eve</td><td>Sofia</td></tr>
  <tr><td>Nick</td><td>Varna</td></tr>
  <tr><td>Didi</td><td>Ruse</td></tr>
  <tr><td>Tedy</td><td>Varna</td></tr>
</table>
<button onclick="colorizeRows()">Colorize</button>
```



Solution: Colorize Table Rows

```
function colorizeRows() {  
    let rows = document.  
        querySelectorAll("table tr");  
    let index = 0;  
    for (let row of rows) {  
        index++;  
        if (index % 2 == 0)  
            row.style.background = "teal";  
    }  
}
```

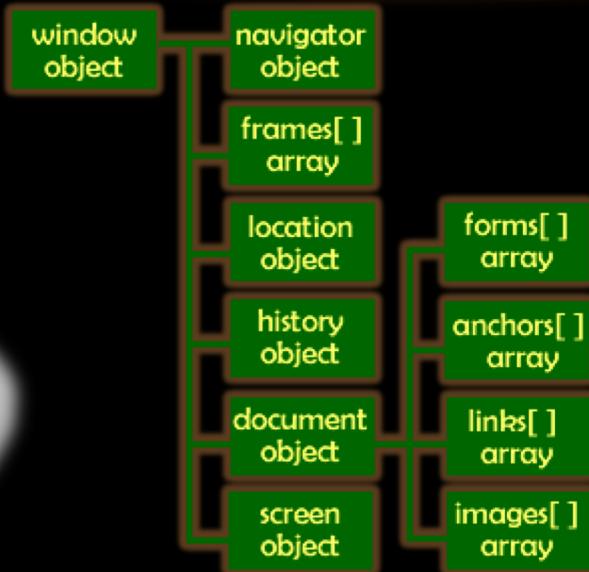


Check your solution here: <https://judge.softuni.bg/Contests/328>



Practice: Document Object Model

Live Exercises in Class (Lab)



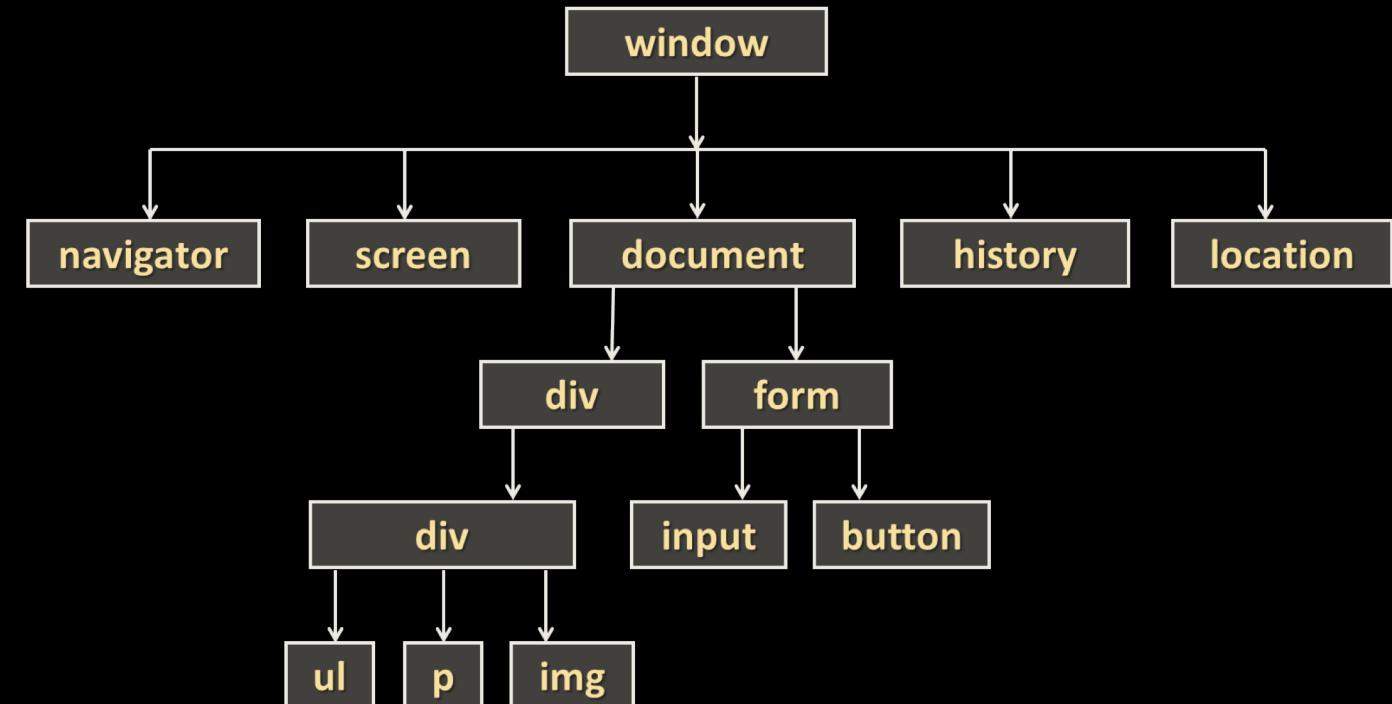
Browser Object Model (BOM)

The Built-In Browser Objects

Browser Object Model (BOM)

- Browsers expose some objects like **window**, **screen**, **navigator**, **history**, **location**, **document**, ...

```
console.dir(window);
console.dir(navigator);
console.dir(screen);
console.dir(location);
console.dir(history);
console.dir(document);
```



Playing with BOM

```
alert(window.navigator.userAgent);
```

This page says:

Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/
537.36

OK

```
console.log(navigator.language);
// en-US
```

```
console.log(screen.width + " x " + screen.height);
// 1920 x 1080
```

```
document.location = "https://softuni.bg";
```

```
history.back();
```

SetInterval() / ClearInterval()

- In JS we can start / stop timers (intervals)

```
let intervalID = setInterval(  
    function() {  
        console.log("1 sec. passed");  
    },  
    1000  
); // Delay = 1000 ms = 1 second
```

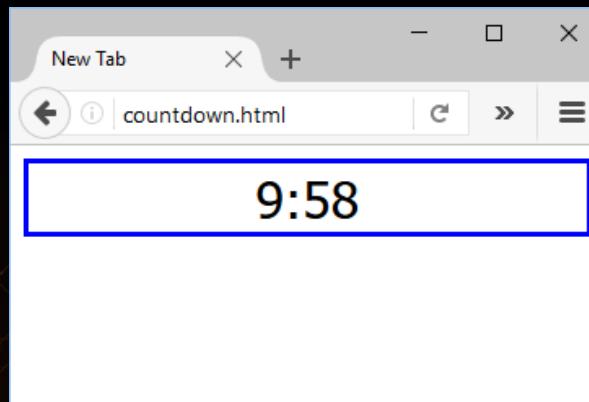


- Remove (cancel) existing timer

```
clearInterval(intervalID); // Stop the timer
```

Problem: Countdown Timer

- An HTML page contains an empty text **input** field
 - Implement the missing JS function **countdown(startTime)**
 - It takes one **number** parameter, the starting time in seconds
 - Each second, it ticks down and displays the current time in the input field



```
<input type="text" id="time"
       style="border:3px solid blue;
              text-align:center; font-size:2em;"  

       disabled="true"/>  

<script>window.onload = function() {
  countdown(600); }</script>
```

Solution: Countdown Timer

```
function countdown(startTime) {  
    let time = startTime;  
    let box = document.getElementById('time');  
    let intervalID = setInterval(decrement, 1000);  
  
    function decrement() {  
        time--;  
        box.value = Math.trunc(time / 60) +  
                    ':' + ("0" + (time % 60)).slice(-2);  
    }  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/328>

Problem: Extract Parenthesis

- Extract all **parenthesized text** from a target paragraph
 - Your function will receive an element ID to parse
 - Return the result as string, joined by " ; "

```
<html>
  <head></head>
  <body>
    <p id="content">
      Rakiya (Bulgarian brandy) is home-made liquor
      (alcoholic drink). It can be made of grapes,
      plums or other fruits (even apples).
    </p>
  </body>
</html>
```

```
  >>let text = extract("content")|
```



Bulgarian brandy;
alcoholic drink;
even apples

Problem: Extract Parenthesis (2)

■ Sample HTML

```
<p id="content">  
    Rakiya (Bulgarian brandy) is home-made liquor  
    (alcoholic drink). It can be made of grapes, plums or  
    other fruits (even apples).  
</p>  
<p id="holder">  
    Lorem ipsum dolor sit amet, (consectetur adipiscing  
    elit), sed do eiusmod (tempor) incididunt ut labore  
    (et dolore magna) aliqua.  
</p>
```

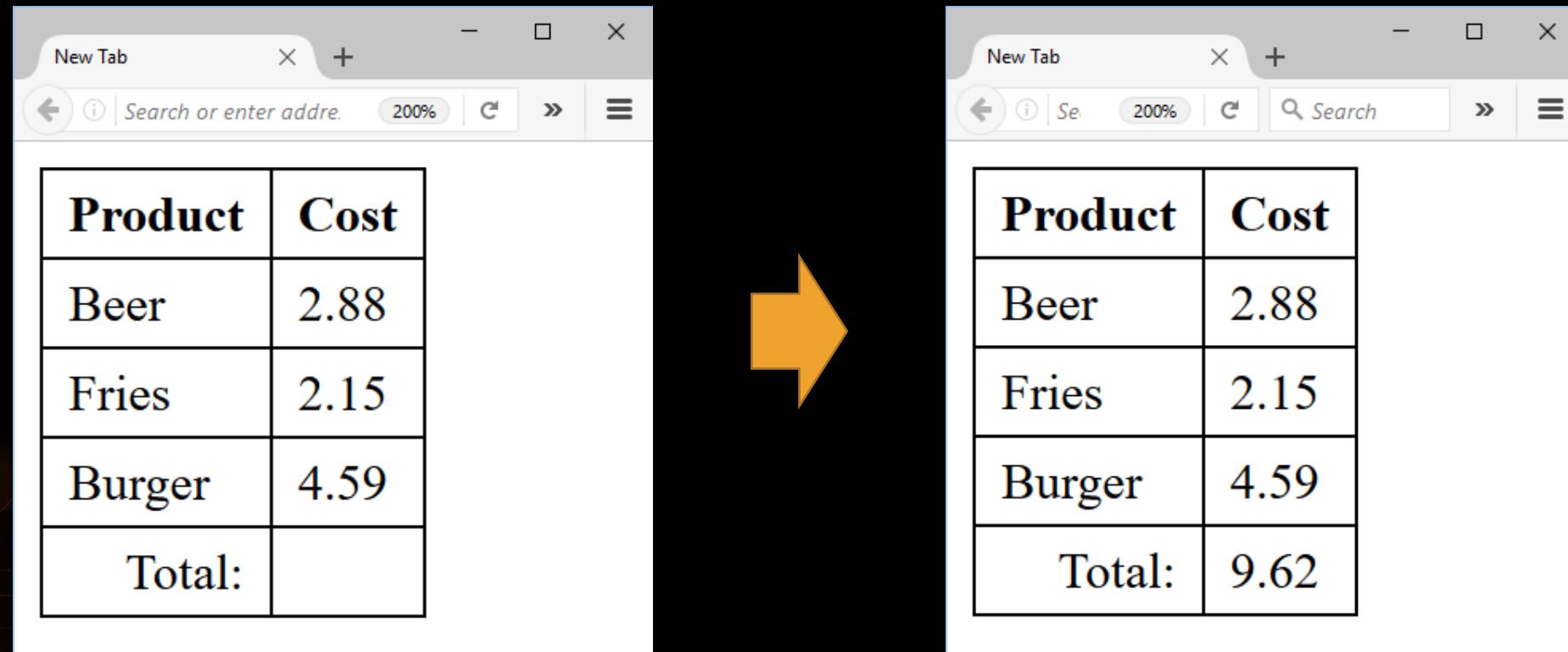
Solution: Extract Parenthesis

```
function extract(elementId) {  
    let para = document.getElementById(elementId).textContent;  
    let pattern = /\((([^)]+))\)/g;  
    let result = [];  
  
    let match = pattern.exec(para);  
    while(match) {  
        result.push(match[1]);  
        match = pattern.exec(para);  
    }  
  
    return result.join('; ');  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/328>

Problem: Sum Table

- Find the first table and sum all values in the last column
- Display the result inside element with ID "sum"



The diagram illustrates a web development task. On the left, a browser window shows a table with four rows: Beer (cost 2.88), Fries (cost 2.15), Burger (cost 4.59), and a blank row labeled 'Total:'. A large orange arrow points from this state to the right. On the right, another browser window shows the same table structure, but the last row now contains 'Total:' in the first column and '9.62' in the second column, indicating the sum of the previous values.

Product	Cost
Beer	2.88
Fries	2.15
Burger	4.59
Total:	

Product	Cost
Beer	2.88
Fries	2.15
Burger	4.59
Total:	9.62

Problem: Sum Table (2)

- Sample HTML

```
<table>
  <tbody>
    <tr><th>Product</th><th>Cost</th></tr>
    <tr><td>Beer</td>      <td>2.88</td></tr>
    <tr><td>Fries</td>      <td>2.15</td></tr>
    <tr><td>Burger</td>      <td>4.59</td></tr>
    <tr><td>Total:</td>      <td id="sum"></td></tr>
  </tbody>
</table>
<button onclick="sum()">Sum</button>
```

Solution: Sum Table

```
function sum() {  
    let table = document.querySelectorAll("table tr");  
    let total = 0;  
    for (let i = 1; i < table.length; i++) {  
        let cols = table[i].children;  
        let cost = cols[cols.length - 1].textContent;  
        total += Number(cost);  
    }  
    document.getElementById("sum").textContent = total;  
}
```



Practice: Document Object Model

Live Exercises in Class (Lab)

Summary

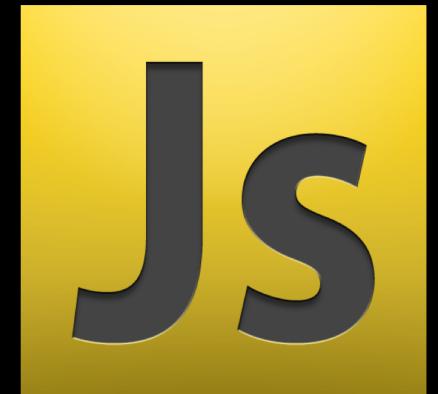
- Browsers keep elements as DOM tree
- Finding / modifying DOM elements:

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



- Accessing content:

```
let text = element.textContent;
element.textContent = "Hello, DOM!"
```



Document Object Model



Questions?

SUPERHOSTING.BG

INDEAVR
Serving the high achievers



License

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



Free Trainings @ Software University

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

