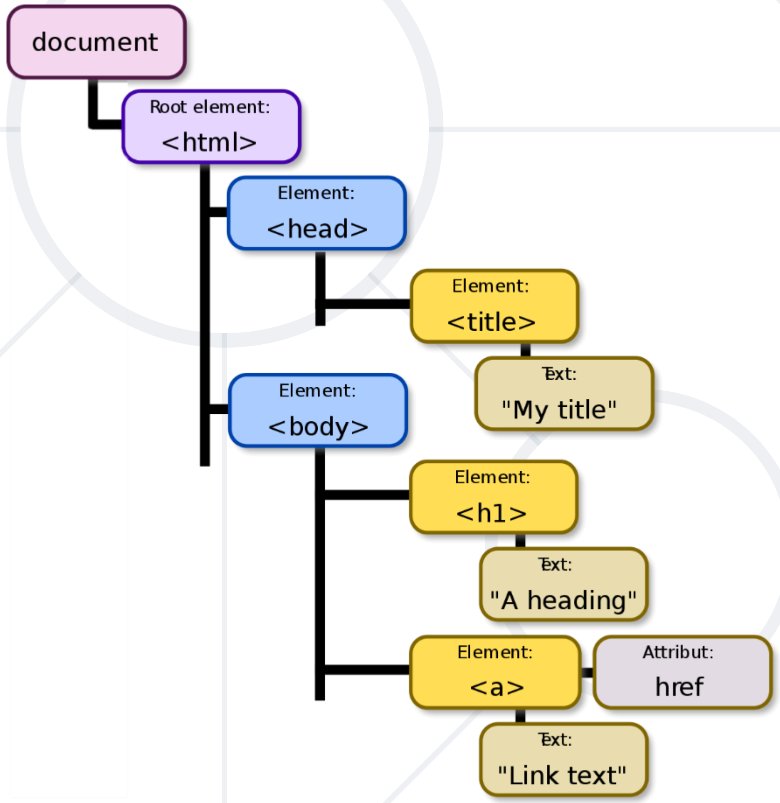
DOM Manipulation

Create / Delete DOM Elements, Handle Browser Events



SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

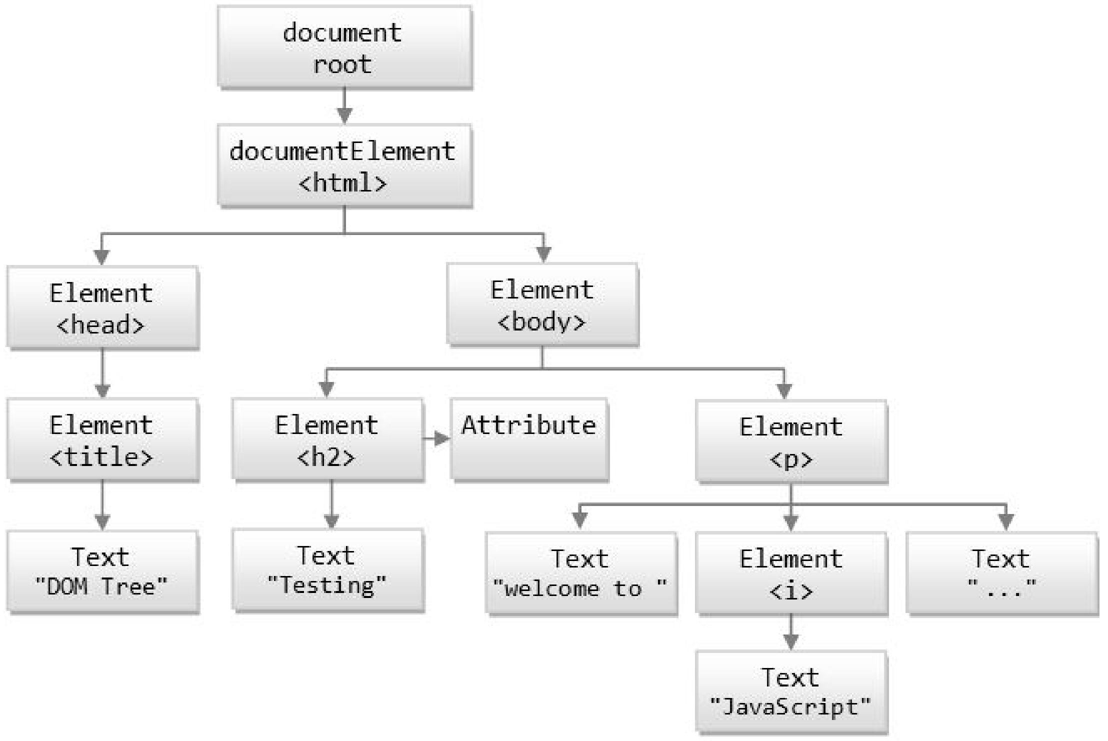
Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#JSCORE



DOM Manipulation

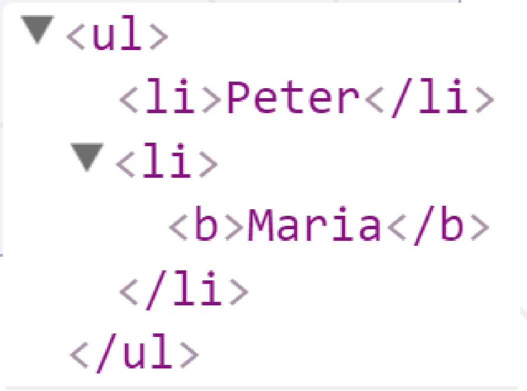
Modify the DOM Tree

Creating New DOM Elements

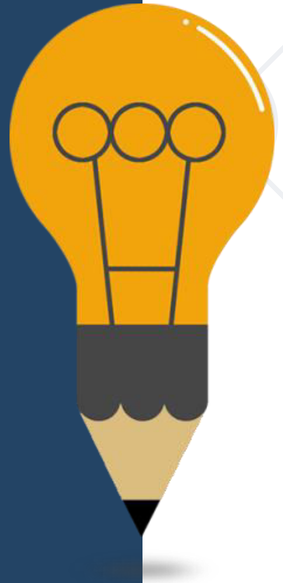
- HTML elements are created with **`document.createElement`**
  - This is called a **Factory Pattern**
- Variables holding HTML elements are **live**:
  - If you **modify** the contents of the variable, the DOM is **updated**
  - If you **insert** it somewhere in the DOM, the original is **moved**
-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

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);
```

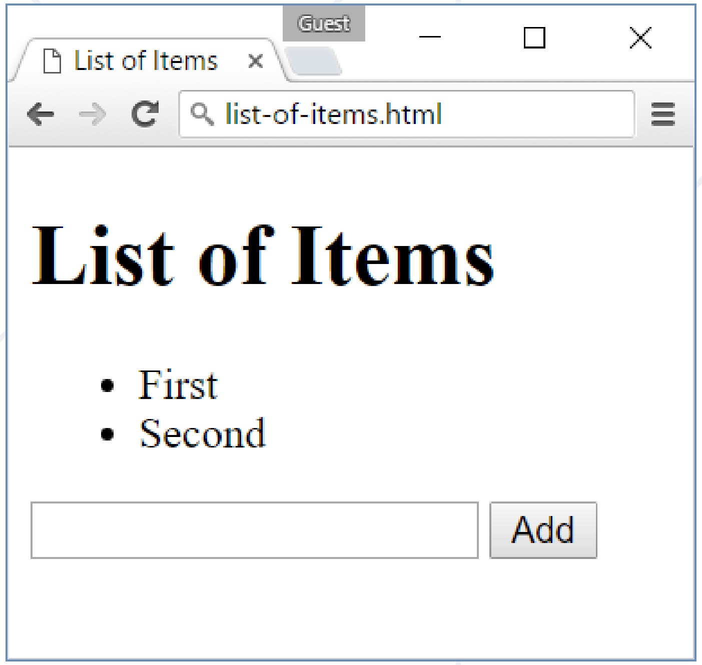


```
▼<ul>
   <li>Peter</li>
  ▼<li>
     <b>Maria</b>
   </li>
 </ul>
```



Problem: List of Items

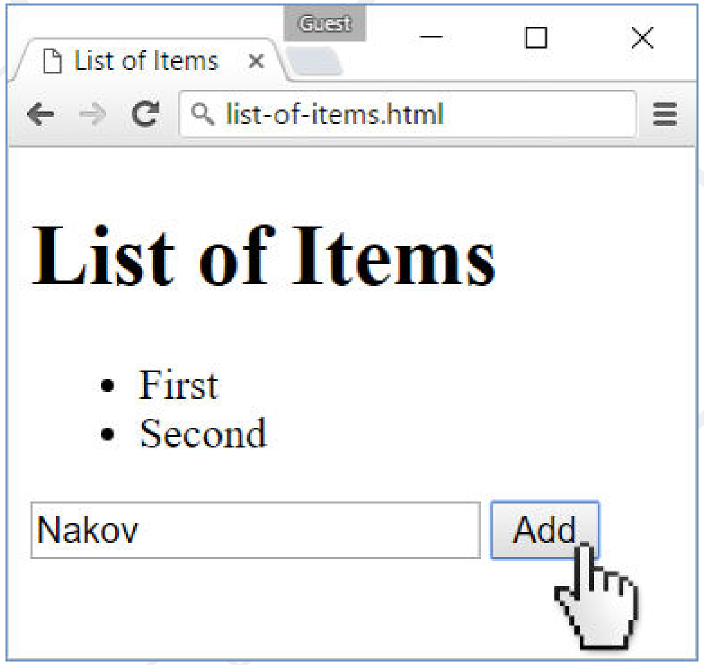
- Create a HTML page holding a **list of items** + **text box** + **button** for adding more items to the list
  - Write a **JS function** to append the specified text to the list



List of Items

- First
- Second

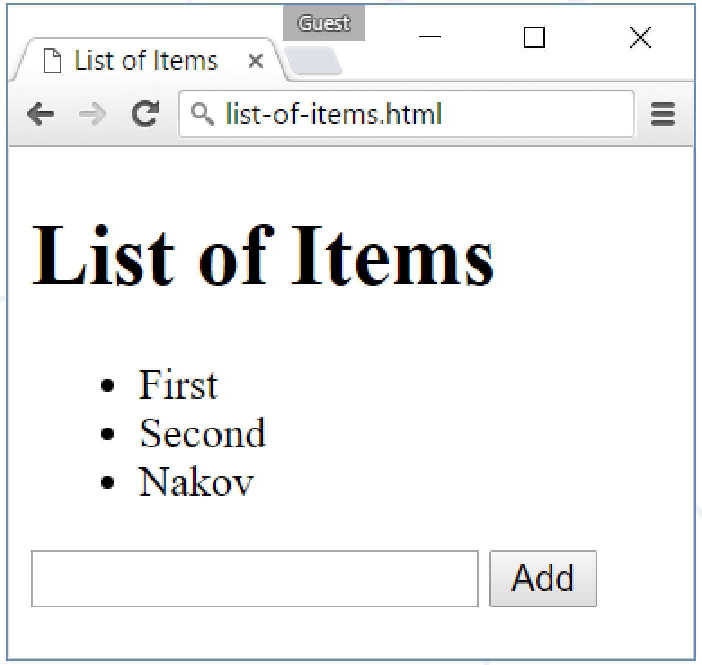
Add



List of Items

- First
- Second

Nakov Add



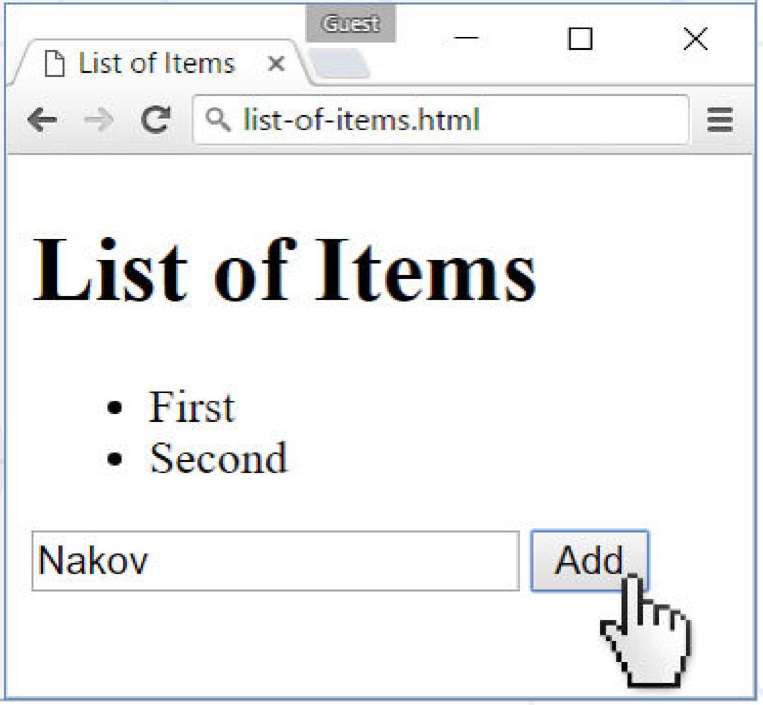
List of Items

- First
- Second
- Nakov

Add

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>
```



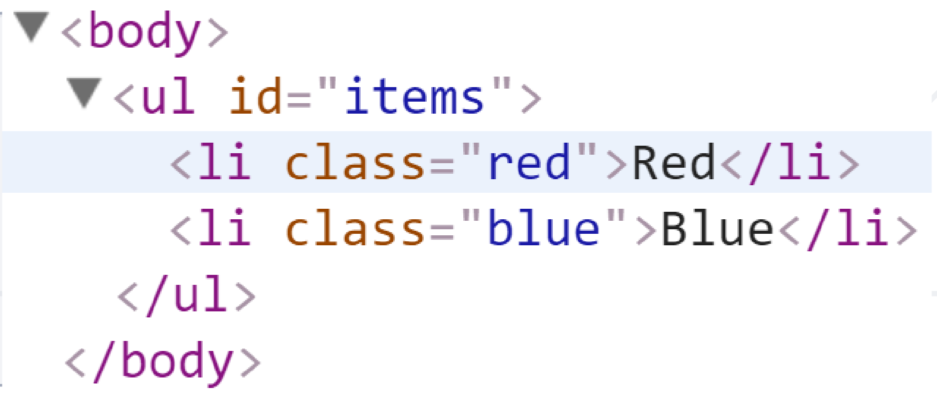
Solution: List of Items

```
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/328

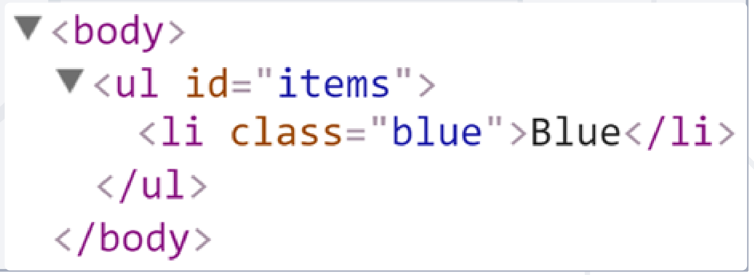
Deleting DOM Elements

```html
<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>
```

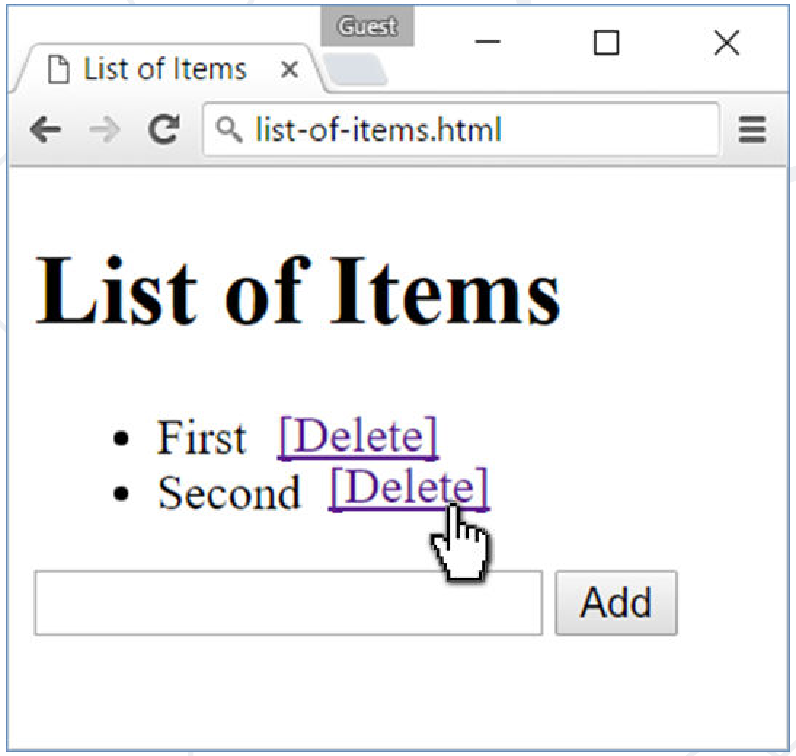
```javascript
let redElements =
  document.querySelectorAll("#items li.red");
redElements.forEach(li => {
  li.parentNode.removeChild(li);
});
```



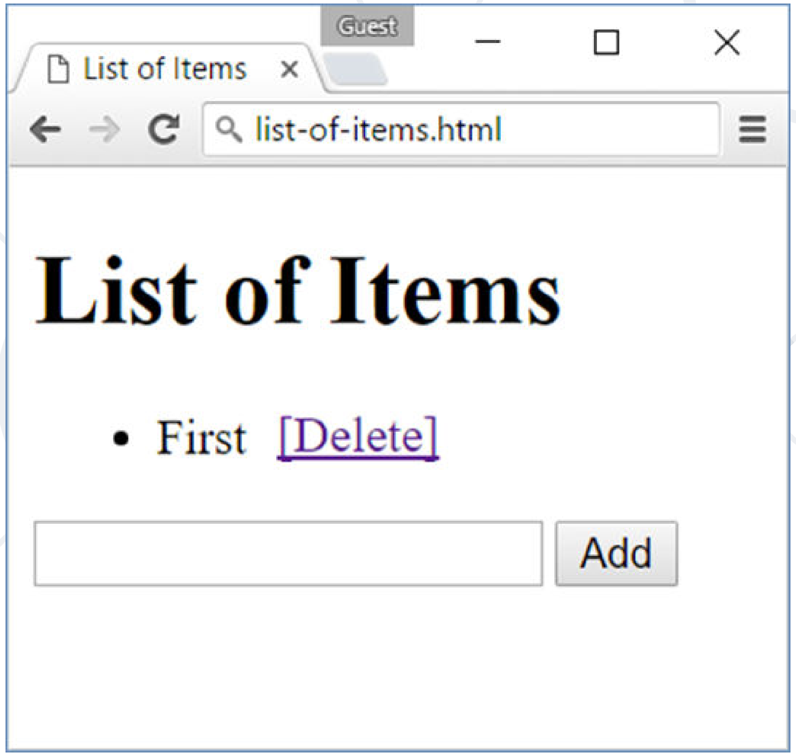
```
▼<body>
  ▼<ul id="items">
     <li class="blue">Blue</li>
   </ul>
 </body>
```

Problem: Add / Delete Items

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


List of Items

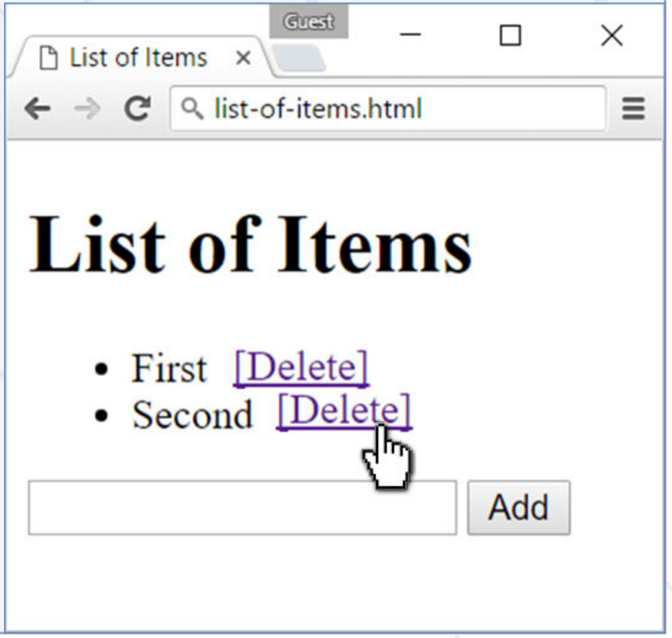
- First [Delete]
- Second [Delete]


List of Items

- First [Delete]

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>
```



List of Items

list-of-items.html

List of Items

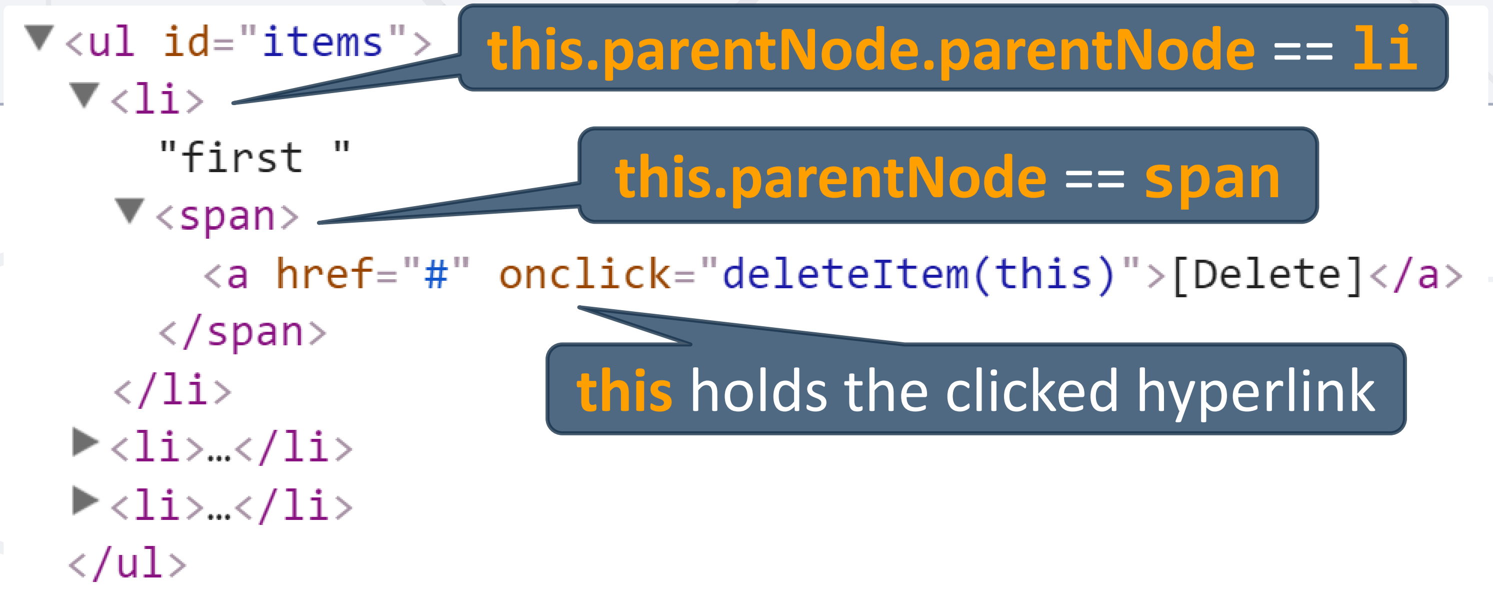
- First [Delete]
- Second [Delete]

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);
  }
}
```



```
▼<ul id="items">
  ▼<li>
      "first "
    ▼<span>
        <a href="#" onclick="deleteItem(this)">[Delete]</a>
      </span>
    </li>
  ▶<li>…</li>
  ▶<li>…</li>
  </ul>
```

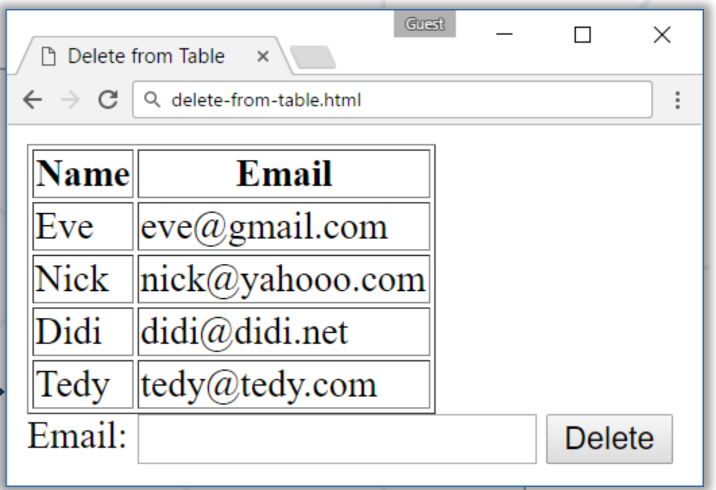
this.parentNode.parentNode == li
this.parentNode == span
this holds the clicked hyperlink

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

Problem: Delete from Table

SoftUni Foundation

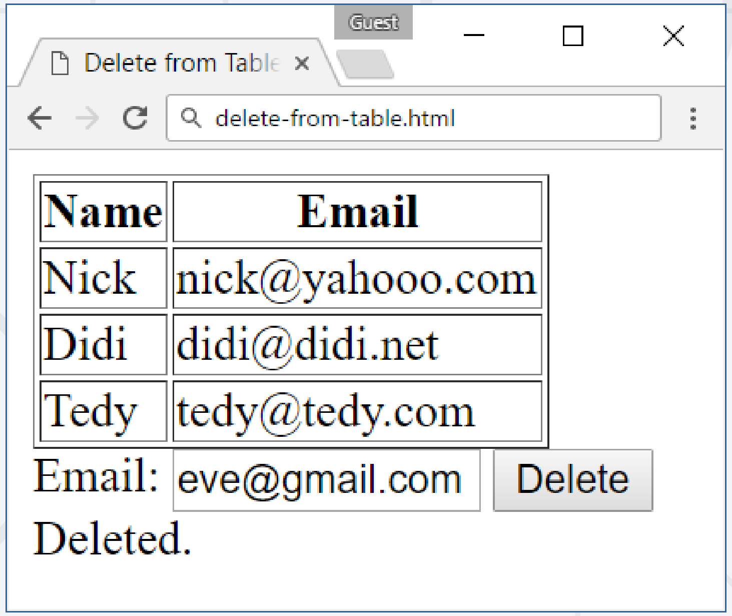
```
<table border="1" id="customers">
 <tr><th>Name</th><th>Email</th></tr>
 <tr><td>Eve</td><td>eve@gmail.com</td></tr>
 <tr><td>Nick</td><td>nick@yahooo.com</td></tr>
 <tr><td>Didi</td><td>didi@didi.net</td></tr>
 <tr><td>Tedy</td><td>tedy@tedy.com</td></tr>
</table>
Email: <input type="text" name="email" />
<button onclick="deleteByEmail()">Delete</button>
<div id="result" />
```



Name	Email
Eve	eve@gmail.com
Nick	nick@yahooo.com
Didi	didi@didi.net
Tedy	tedy@tedy.com

Solution: Delete from Table

```
function deleteByEmail() {
  let email = document.getElementsByName("email")[0].value;
  let secondColumn = document.querySelectorAll(
    "#customers tr td:nth-child(2)");
  for (let td of secondColumn)
    if (td.textContent == email) {
      let row = td.parentNode;
      row.parentNode.removeChild(row);
      document.getElementById('result').
        textContent = "Deleted.";
      return;
    }
  document.getElementById('result').textContent = "Not found.";
}
```



Name	Email
Nick	nick@yahooo.com
Didi	didi@didi.net
Tedy	tedy@tedy.com

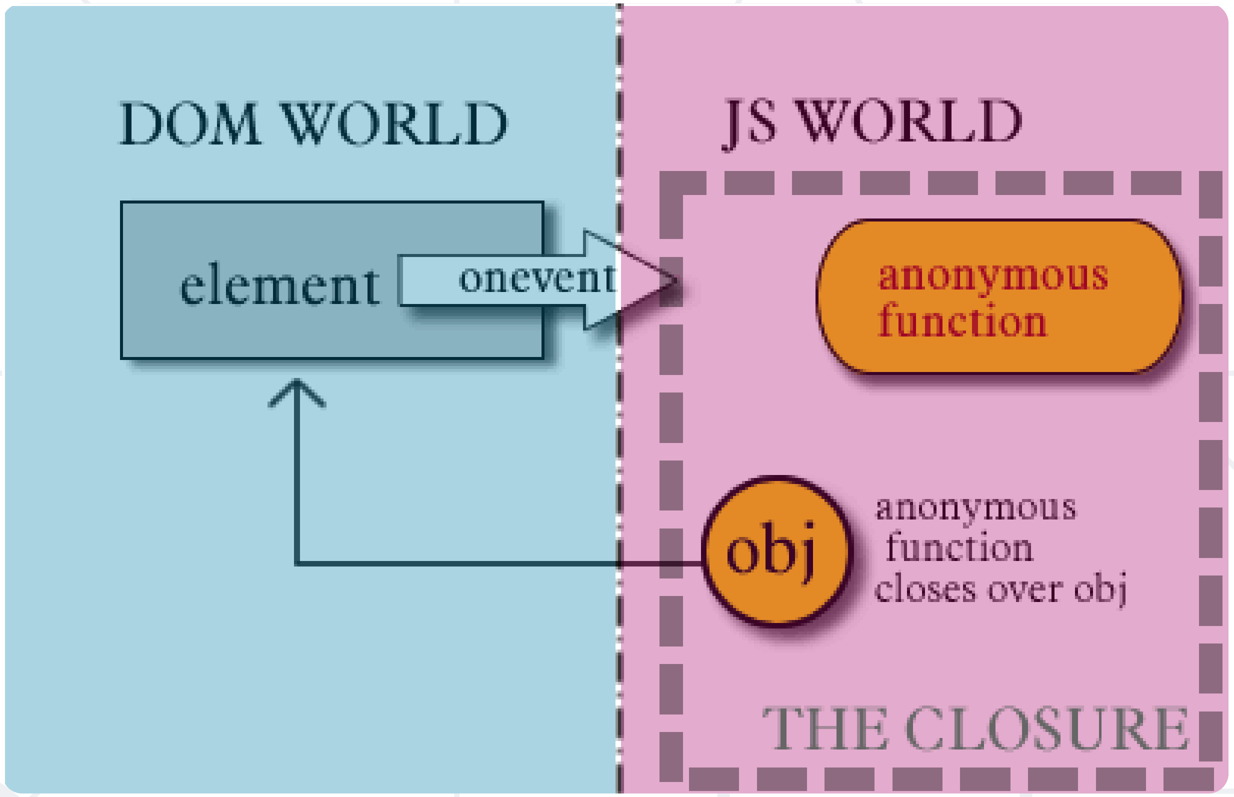
Email: eve@gmail.com Delete

Deleted.

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

Practice: DOM and Events

Live Exercises in Class (Lab)



Handling Events

Browser Events and DOM Events

Handling Events in JS

- Browsers send **events** to notify the JS code of interesting things that have taken place

```
<div id='text'>Some text</div>
```

```
let div = document.getElementById('text');
div.onmouseover = function(event) {
  event.target.style.border = "3px solid green";
}
div.onmouseout = function() {
  this.style.border = ""; // this === event.target
}
```

Event Types in DOM API

- **Mouse** events

```
click
mouseover
mouseout
mousedown
mouseup
```

- **Touch** events

```
touchstart
touchend
touchmove
touchcancel
```

- **DOM / UI** events

```
load (finished loading)
unload (exit from page)
resize (window resized)
dragstart / drop
```

- **Keyboard** events

```
keydown
keypress (emit char)
keyup
```

- **Focus** events

```
focus (got focus)
blur (lost focus)
```

- **Form** events

```
input (value changed)
change (change + leave)
submit (form sent)
reset (form reset)
```

Learn more at https://developer.mozilla.org/docs/Web/Events

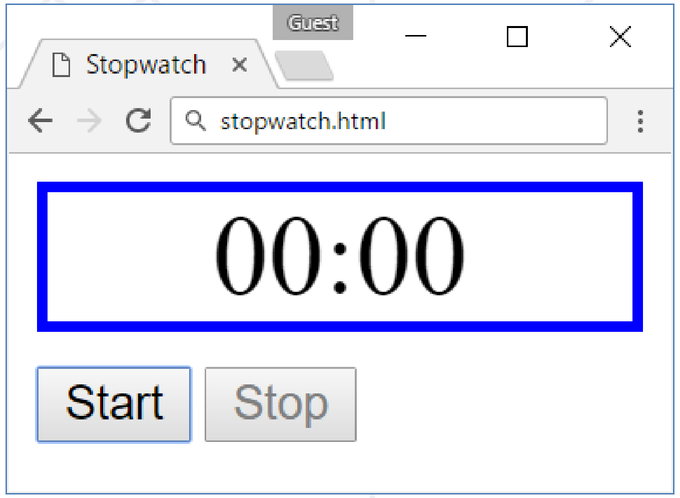
Add / Remove Event Handler

```
let textbox = document.createElement('input');
textbox.type = 'text';
textbox.value = "I am a text box";
document.body.appendChild(textbox);

textbox.addEventListener('focus', focusHandler);

function focusHandler(event) {
  textbox.value = "Event handler removed";
  textbox.removeEventListener('focus', focusHandler);
}
```

Subscribe to events like this, don't use **onclick** / **onfocus**

Problem: Stopwatch

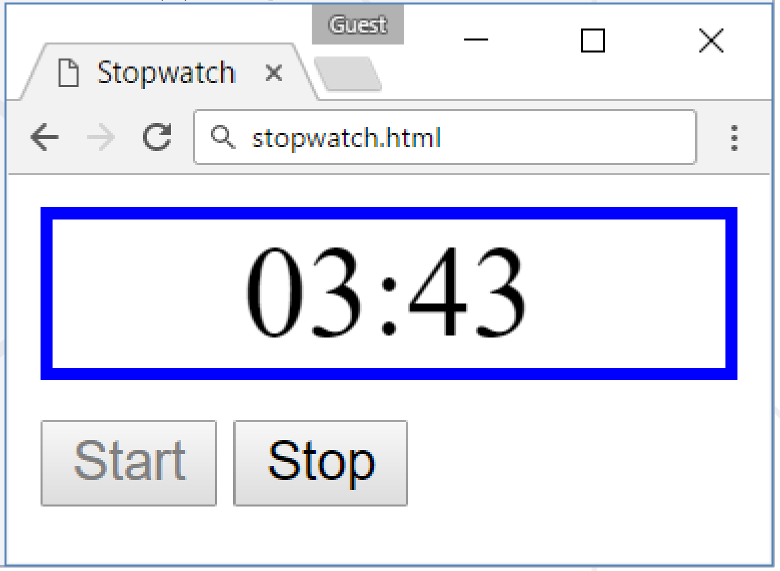
- A HTML page holds **time-box** + **[Start]** + **[Stop]** buttons
  - Implement the missing JS function **stopwatch()**
    - Increase the time at each second
    - Disable / enable buttons



```
<div id="time" style="border:3px
   solid blue; text-align:center;
   font-size:2em; margin-
   bottom:10px">00:00</div>
<button id="startBtn">Start</button>
<button id="stopBtn"
   disabled="true">Stop</button>
<script>window.onload = function() {
   stopwatch(); }</script>
```

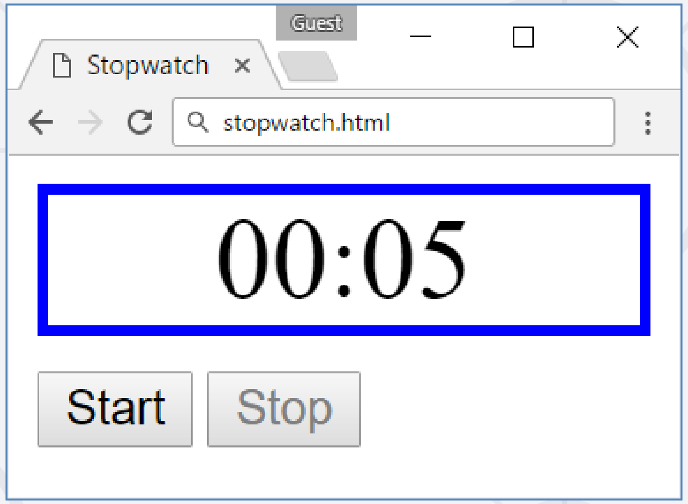
Solution: Stopwatch

```
function stopwatch() {
  let time, intervalID;
  let startBtn = document.getElementById('startBtn');
  let stopBtn = document.getElementById('stopBtn');

  startBtn.addEventListener('click', function() {
    time = -1;
    incrementTime();
    intervalID = setInterval(
      incrementTime, 1000);
    startBtn.disabled = true;
    stopBtn.disabled = false;
  });
```



Solution: Stopwatch (2)

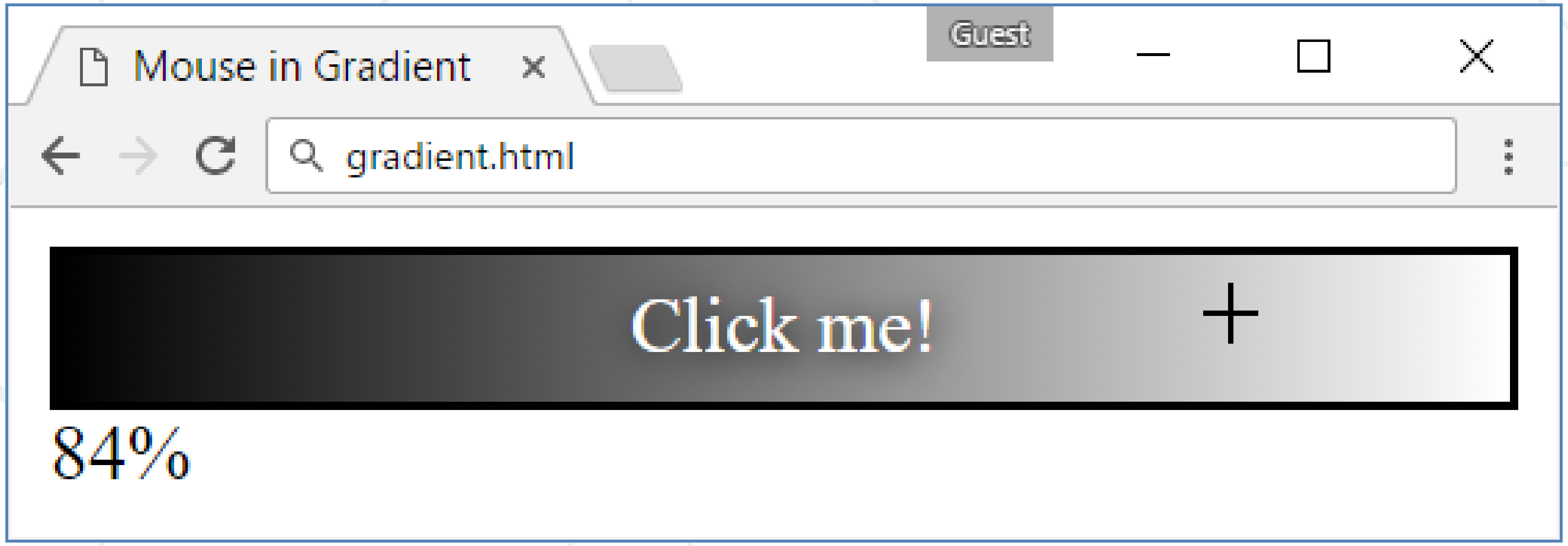
```
  stopBtn.addEventListener('click', function() {
    clearInterval(intervalID);
    startBtn.disabled = false;
    stopBtn.disabled = true;
  });

  function incrementTime() {
    time++;
    document.getElementById('time').textContent =
      ("0" + Math.trunc(time / 60)).slice(-2) +
      ':' + ("0" + (time % 60)).slice(-2);
  }
}
```



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

Problem: Mouse in Gradient

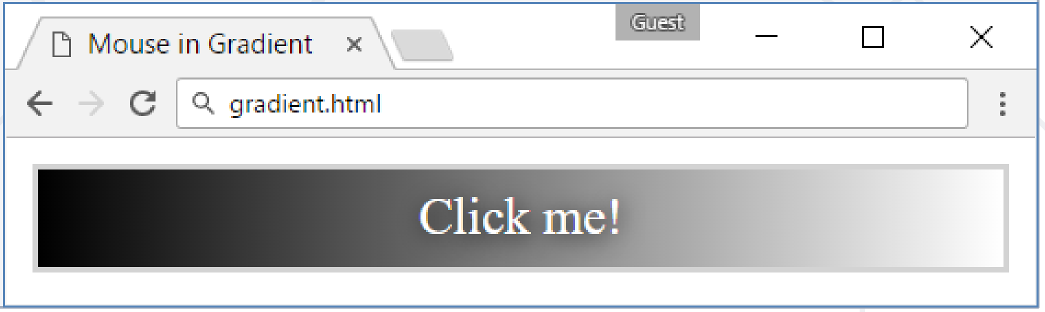
- A HTML page holds **linear gradient** box
  - Moving the mouse should show **percentage** [0% … 100%], depending on the **location of mouse**
  - Left side → **0%**; middle → **50%**; right side → **100%**



Click me!
84%

Problem: Mouse in Gradient – HTML

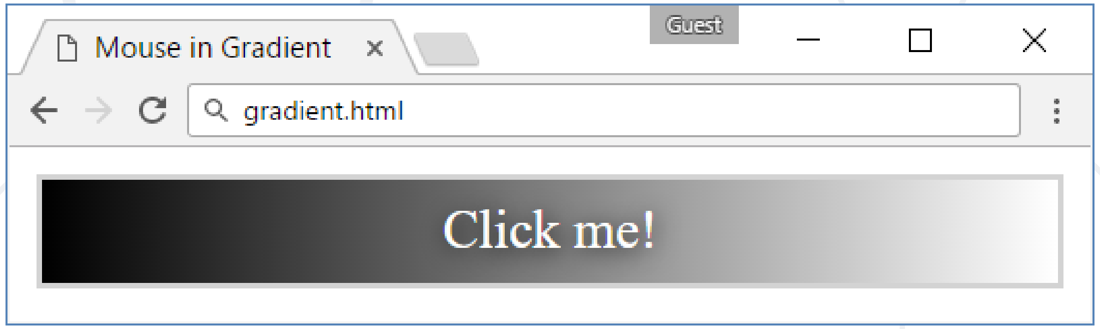
```
<html>
<head>
  <title>Mouse in Gradient</title>
  <link rel="stylesheet" href="gradient.css" />
  <script src="gradient.js"></script>
</head>
<body onload="attachGradientEvents()">
  <div id="gradient-box">
    <div id="gradient">Click me!</div>
  </div>
  <div id="result"></div>
</body>
</html>
```



Problem: Mouse in Gradient – CSS

```css
#gradient-box {
  width: 300px;
  border: 2px solid lightgrey;
}
#gradient-box:hover {
  border: 2px solid black;
}
#gradient {
  height: 30px;
  color: white;
  text-shadow:
    1px 1px 10px black;
```

```css
  text-align: center;
  line-height: 30px;
  background:
    linear-gradient(
    to right, black, white);
  cursor: crosshair;
}
```



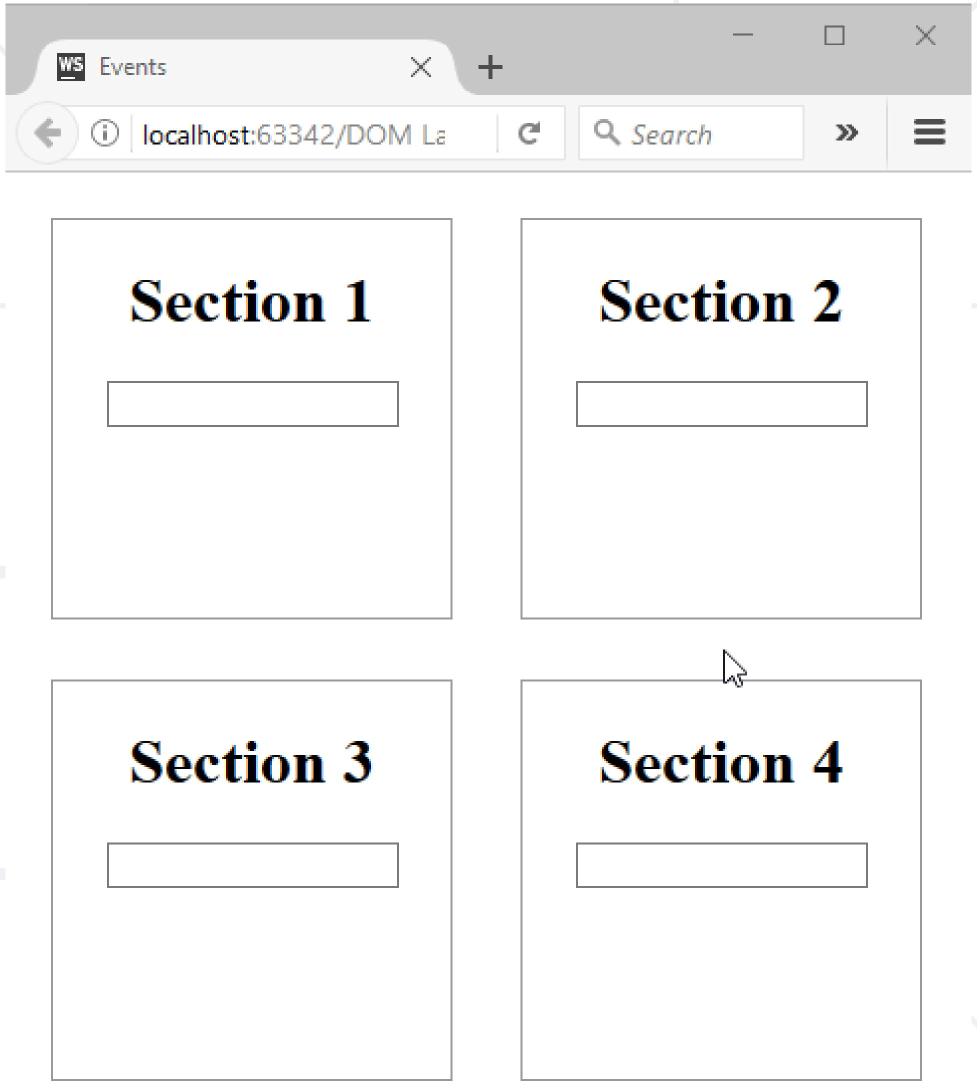
Solution: Mouse in Gradient

```
function attachGradientEvents() {
  let gradient = document.getElementById('gradient');
  gradient.addEventListener('mousemove', gradientMove);
  gradient.addEventListener('mouseout', gradientOut);

  function gradientMove(event) {
    let power = event.offsetX / (event.target.clientWidth - 1);
    power = Math.trunc(power * 100);
    document.getElementById('result').textContent = power + "%";
  }

  function gradientOut(event) {
    document.getElementById('result').textContent = "";
  }
};
```

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

Problem: Highlight Active

- A webpage contains **multiple** input boxes inside divs
- Apply styling to the div that holds the **focused** input box
  - Set class "**focus**" for active
  - **Remove** class **attribute** from inactive



Problem: Highlight Active (2)

```html
<!DOCTYPE html><html lang="en">
<head>
  <meta charset="UTF-8"><title>Focus</title>
  <link rel="stylesheet" href="focus.css" />
  <script src="focus.js"></script>
</head>
<body onload="focus()">
  <div>
    <div><h1>Section 1</h1><input type="text"/></div>
    <div><h1>Section 2</h1><input type="text"/></div>
    <div><h1>Section 3</h1><input type="text"/></div>
    <div><h1>Section 4</h1><input type="text"/></div>
  </div>
</body>
</html>
```

Problem: Highlight Active (3)

- Place all project files in the **same folder**
- Listen for **focus** and **blur** events

focus.js

```
function focus() {
  // TODO
}
```

focus.css

```
div {
  width: 470px;
}
div div {
  text-align: center;
  display: inline-block;
  width: 200px;
  height: 200px;
  margin: 15px;
  border: 1px solid #999;
}
.focused {
  background: #999999;
}
```

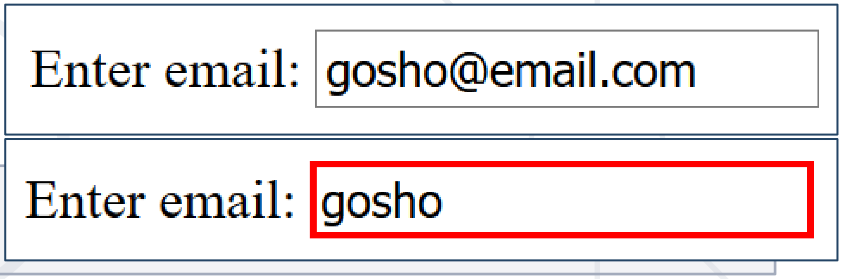
Solution: Highlight Active

```
function focus() {
  let inputs = document.getElementsByTagName('input');
  Array.from(inputs).forEach(i => {
    i.addEventListener('focus', (event) => {
      event.target.parentNode.className = 'focused';
    });
    i.addEventListener('blur', (event) => {
      event.target.parentNode.removeAttribute('class');
    });
  });
}
```

Problem: Dynamic Validation

SoftUni Foundation

- A webpage contains a single email **input** field
- Display **real-time feedback** for user's input
- Valid input format: **<name>@<domain>.<extension>**
  - Only lowercase Latin letters are allowed for **all parts**
- Apply class "**error**" when input is **invalid**



Enter email: gosho@email.com

Enter email: gosho

```
.error { border: 2px solid red; }
```

```
<label for="email">Enter email:</label>
<input id="email" type="text"/>
```

Solution: Dynamic Validation

```
function validate() {
  document.querySelector('input')
    .addEventListener('change', onChange);
  let regex = /^([\w\-.]+)@([a-z]+)(\.[a-z]+)+$/;

  function onChange(event) {
    if (!regex.test(event.target.value))
      event.target.className = 'error';
    else
      event.target.removeAttribute('class');
  }
}
```

Practice: DOM and Events

Live Exercises in Class (Lab)

Summary

- Modifying DOM elements:

```
let menu = document.getElementById('menu');
menu.style.display = 'none';
menu.appendChild(
  document.createElement('hr'));
let link = menu.children[0];
menu.removeChild(link);
```

- Handling events:

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

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

Trainings @ Software University (SoftUni)

SoftUni Foundation

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

SoftUni Foundation

License