JS Advanced: Retake Exam 5 September 2017

Problems for exam preparation for the “JavaScript Advanced” course @ SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/756/.

Problem 4. Modal Dialog (Object Interacting with DOM)

Write a JS **class** that generates and controls a **modal dialog** box. It contains a text message, OK and Cancel buttons and optionally input fields and overlays the rest of the website content. When the user clicks OK, a callback is executed with the values of all input fields. *See the examples for more details.*

The **constructor** of your class needs to take **two** arguments – text message (string) and callback (function). The callback is a function that must be executed when the user clicks OK. Additionally, the class should contain the following functionality:

- Function **`addInput(label, name, type)`** – takes three string arguments and adds an input field to the dialog, with name and type attributes as specified
- Function **`render()`** – **generates the HTML content** and **appends** it to the end of the **document’s body**

The dialog has **two buttons** – OK and Cancel. When OK is clicked, the **callback** of the dialog is executed and the dialog is closed (**removed** from the page). If the dialog had any input fields, their **values** are collected in an **object** with their **names as keys**, and the object is used as a **parameter to the callback**. If Cancel is clicked, the dialog is closed (removed from the page), without any side effect.

A dialog box should have the following HTML structure:

Dialog

```html
<div class="overlay">
  <div class="dialog">
    <p>Dialog, containing message text and input field.</p>
    <label>Name</label>
    <input name="name" type="text">
    <button>OK</button>
    <button>Cancel</button>
  </div>
</div>
```

The paragraph contains the **message**, that is entered trough the constructor. The label and input field have been added using the **`addInput()`** function. If two dialogs are created, they should **not interfere** with each other’s functionality – closing one does not affect the other.

You can use the following HTML skeleton to test your functionality:

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Modal Overlay</title>
  <style>
    .overlay {
      position: fixed;
      left: 0;
      right: 0;
      top: 0;
      bottom: 0;
      background: rgba(0, 0, 0, 0.5);
      text-align: center;
    }

    .dialog {
      border: 1px solid black;
      background: #eeeeee;
      display: inline-block;
      margin: 40vh auto;
      padding: 1em 3em;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
</head>
<body>

<h1>Modal Dialog Overlay</h1>
<button onclick="spawnConfirm()">Create OK/Cancel</button>
<button onclick="spawnInput()">Create Input</button>
<textarea style="resize: none; display: block; width: 300px; height: 200px" disabled
id="output"></textarea>

<script src="solution.js"></script>
<script>
  function spawnConfirm() {
    log('Spawning Confirm dialog');
    let callback = (params) => {
      log('Clicked OK');
    };
    let dialog = new Dialog("Test Dialog, containing OK and Cancel buttons.",
callback);
    dialog.render();
  }

  function spawnInput() {
    log('Spawning Input dialog');
    let callback = (params) => {
      log(`Params: {${Object.keys(params).map(k => k + ': ' + params[k]).join(',
')}}`);
    };
    let dialog = new Dialog("Dialog, containing message text and input field.",
callback);
    dialog.addInput('Name', 'name', 'text');
    dialog.render();
  }

  function log(text) {
    document.getElementById('output').textContent += text + '\n';
  }
</script>
</body>
</html>
```

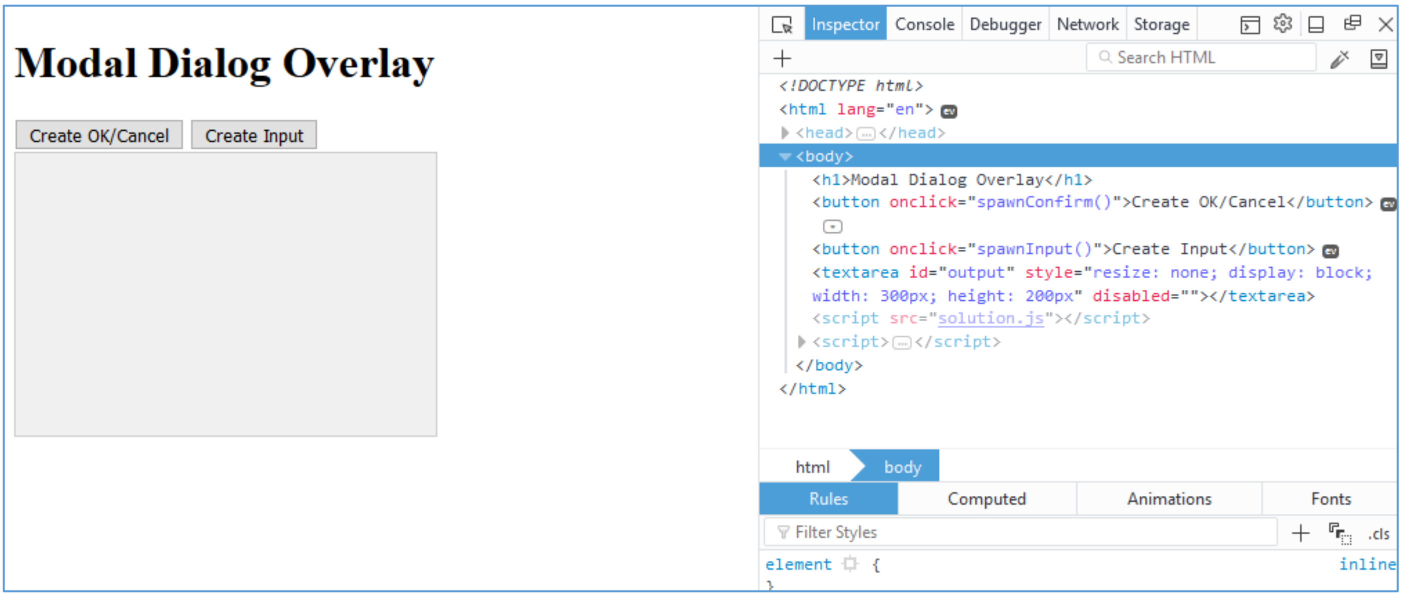
The sample page contains two buttons and a script, that will create two different dialogs that should look like the examples, if your code is correct.

Examples

If we create a dialog and add two inputs to it, one with name “name” and type “text”, and another with name “password” and type “password”, when OK is clicked, the callback must be executed with the following object as parameter:

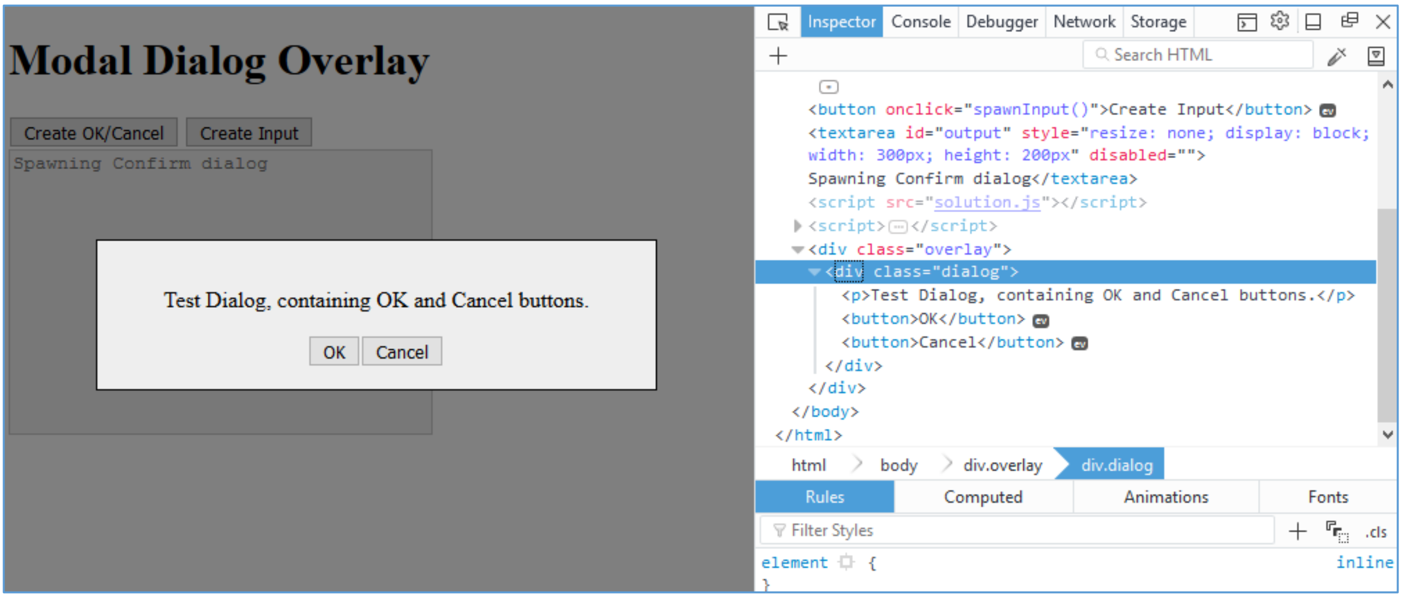
```
{
  name: "pesho",
  password: "123456"
}
```

If there are not inputs, the callback must be executed with an empty object as parameter.



```
<!DOCTYPE html>
<html lang="en">
  <head>…</head>
  <body>
    <h1>Modal Dialog Overlay</h1>
    <button onclick="spawnConfirm()">Create OK/Cancel</button>
    <button onclick="spawnInput()">Create Input</button>
    <textarea id="output" style="resize: none; display: block;
    width: 300px; height: 200px" disabled=""></textarea>
    <script src="solution.js"></script>
    <script>…</script>
  </body>
</html>
```

```
element {
}
```
inline

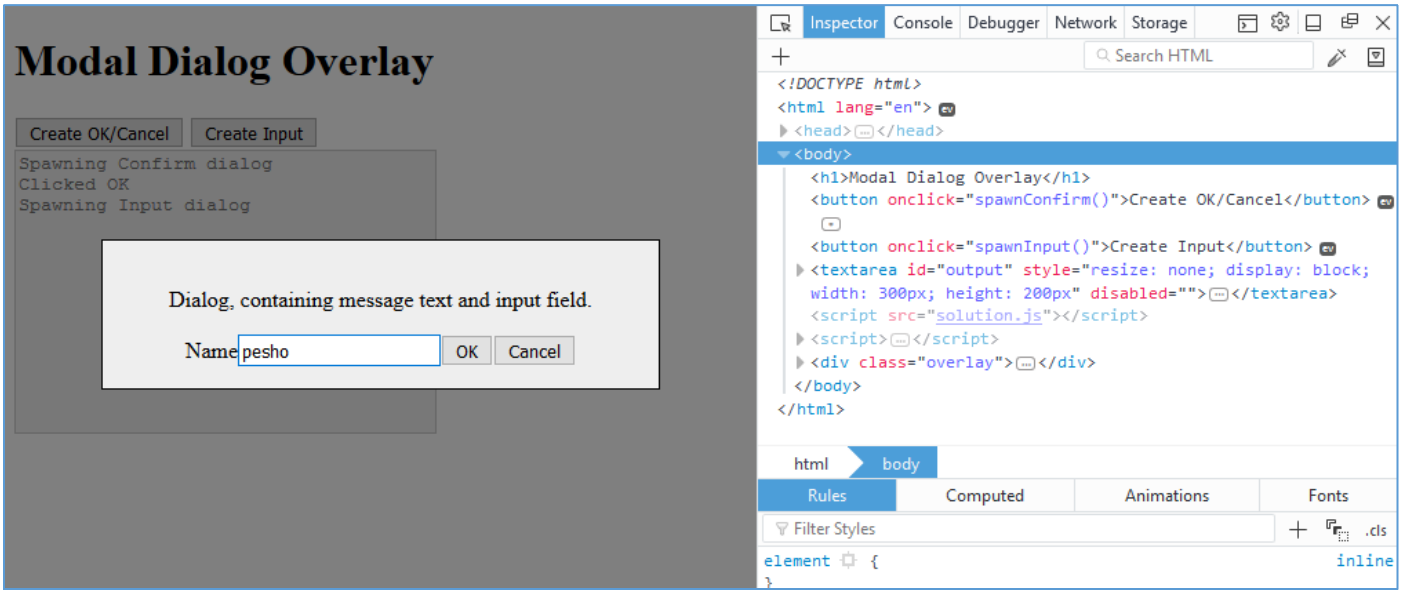


Spawning Confirm dialog

Test Dialog, containing OK and Cancel buttons.

```
    <button onclick="spawnInput()">Create Input</button>
    <textarea id="output" style="resize: none; display: block;
    width: 300px; height: 200px" disabled="">
    Spawning Confirm dialog</textarea>
    <script src="solution.js"></script>
    <script>…</script>
    <div class="overlay">
      <div class="dialog">
        <p>Test Dialog, containing OK and Cancel buttons.</p>
        <button>OK</button>
        <button>Cancel</button>
      </div>
    </div>
  </body>
</html>
```

```
element {
}
```
inline



Modal Dialog Overlay

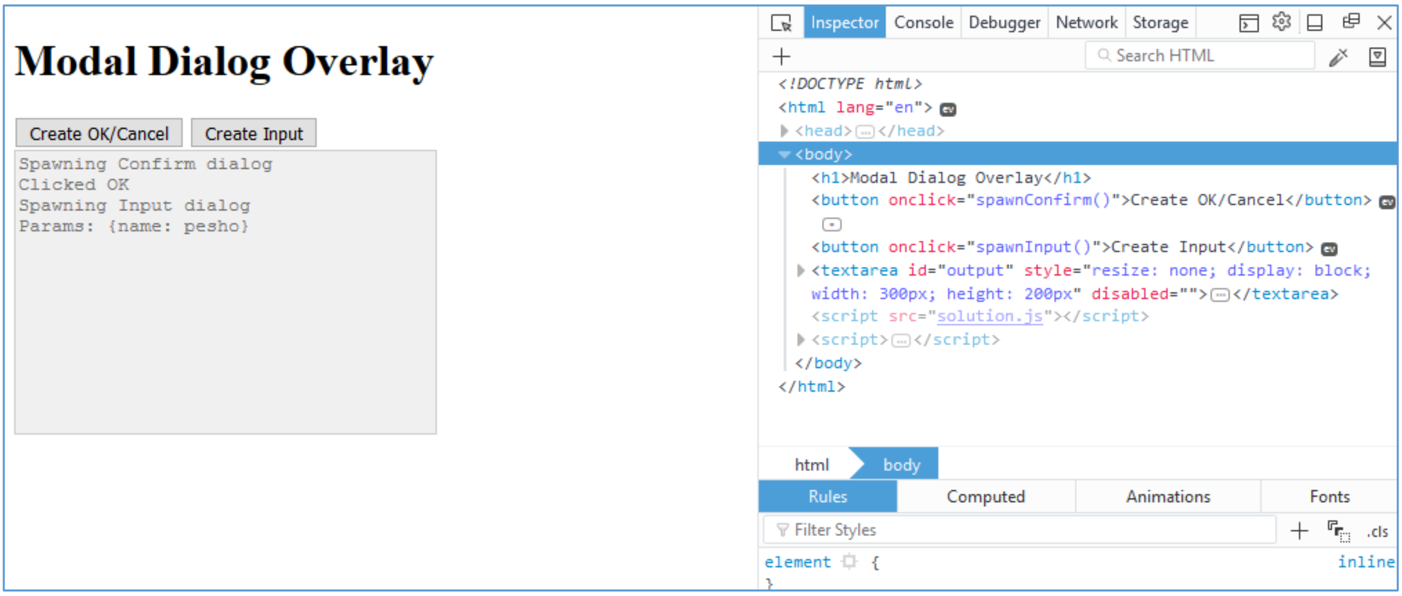
```
Spawning Confirm dialog
Clicked OK
Spawning Input dialog
```

Dialog, containing message text and input field.

Name: pesho

```
<!DOCTYPE html>
<html lang="en"> ev
  <head>…</head>
  <body>
    <h1>Modal Dialog Overlay</h1>
    <button onclick="spawnConfirm()">Create OK/Cancel</button> ev
    
    <button onclick="spawnInput()">Create Input</button> ev
    <textarea id="output" style="resize: none; display: block;
    width: 300px; height: 200px" disabled="">…</textarea>
    <script src="solution.js"></script>
    <script>…</script>
    <div class="overlay">…</div>
  </body>
</html>
```

```
element { inline
}
```



Modal Dialog Overlay

```
Spawning Confirm dialog
Clicked OK
Spawning Input dialog
Params: {name: pesho}
```

```
<!DOCTYPE html>
<html lang="en"> ev
  <head>…</head>
  <body>
    <h1>Modal Dialog Overlay</h1>
    <button onclick="spawnConfirm()">Create OK/Cancel</button> ev
    
    <button onclick="spawnInput()">Create Input</button> ev
    <textarea id="output" style="resize: none; display: block;
    width: 300px; height: 200px" disabled="">…</textarea>
    <script src="solution.js"></script>
    <script>…</script>
  </body>
</html>
```

```
element { inline
}
```