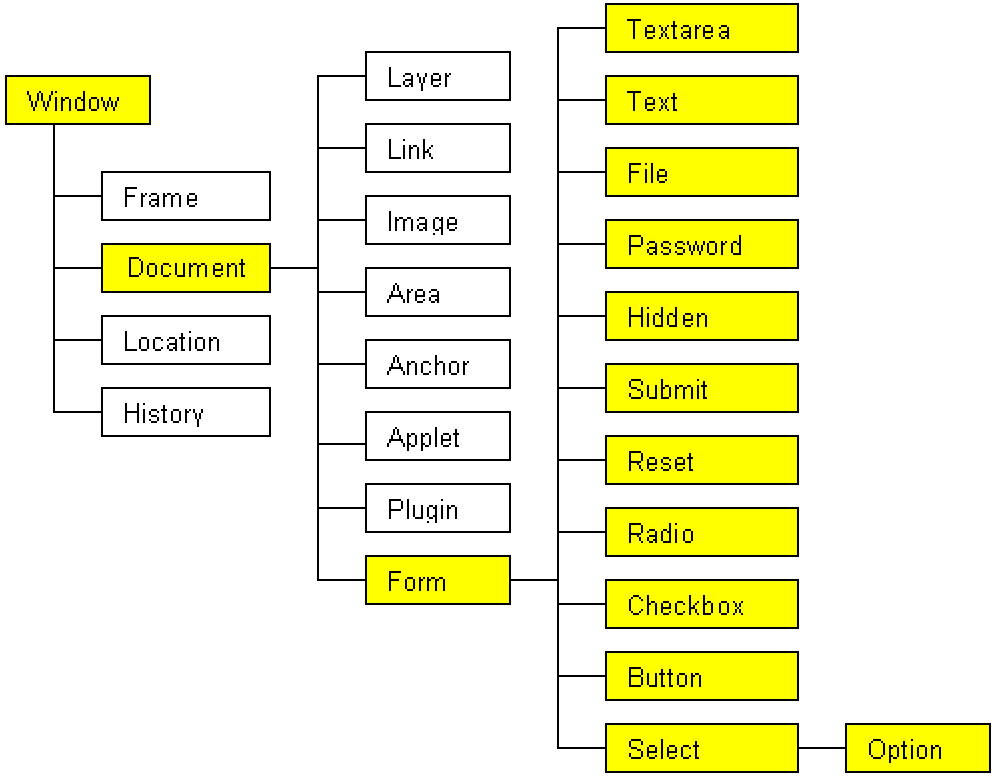
Javascript

PART 7 - FORMS

Forms

DOM forms hierarchy:



Form validation

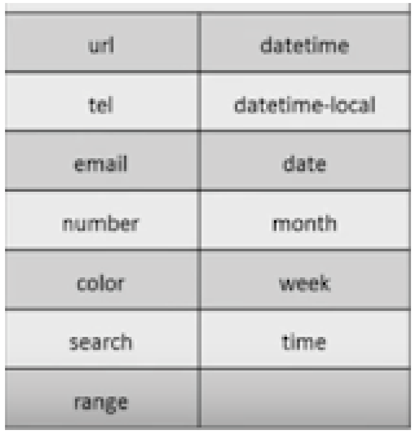
There are three possible ways for validating a form:

1) HTML5 Built in form validation: Better performance, but not as customizable
2) Constraint Validation DOM Methods
3) JavaScript validation

1. HTML built-in validation

Better performance, but not as customizable
It is the method that we have been using during the first term of the course

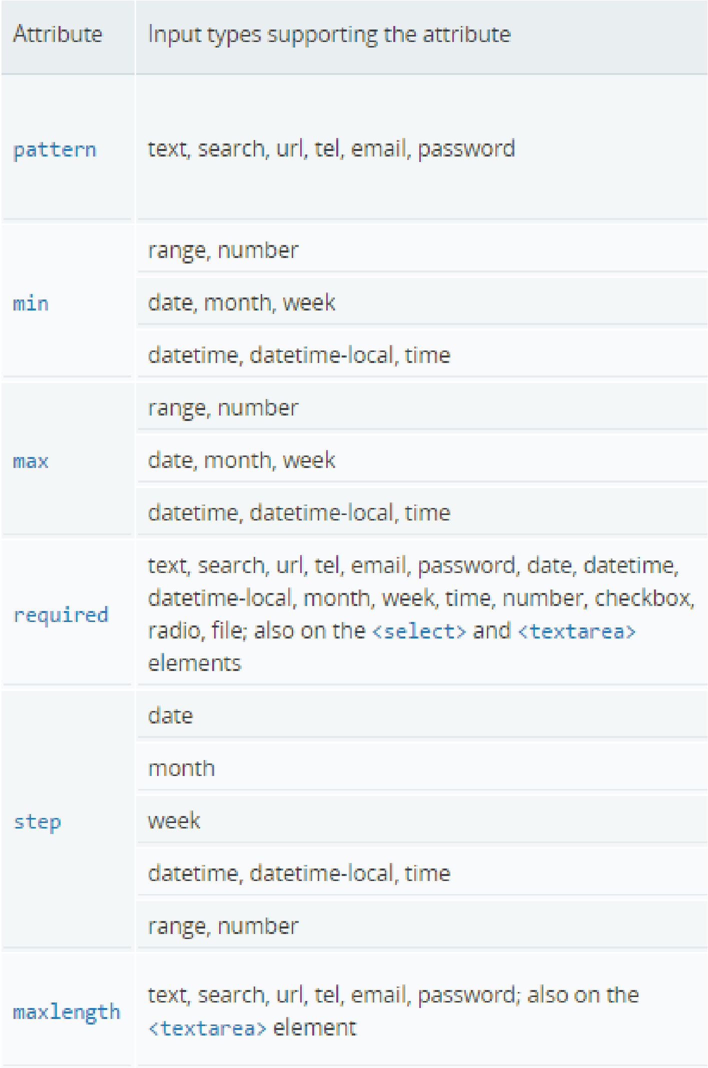
- Values for input types:



url	datetime
tel	datetime-local
email	date
number	month
color	week
search	time
range	

Example: <input type='number' min='2' max='5'>

Validation-related attributes:

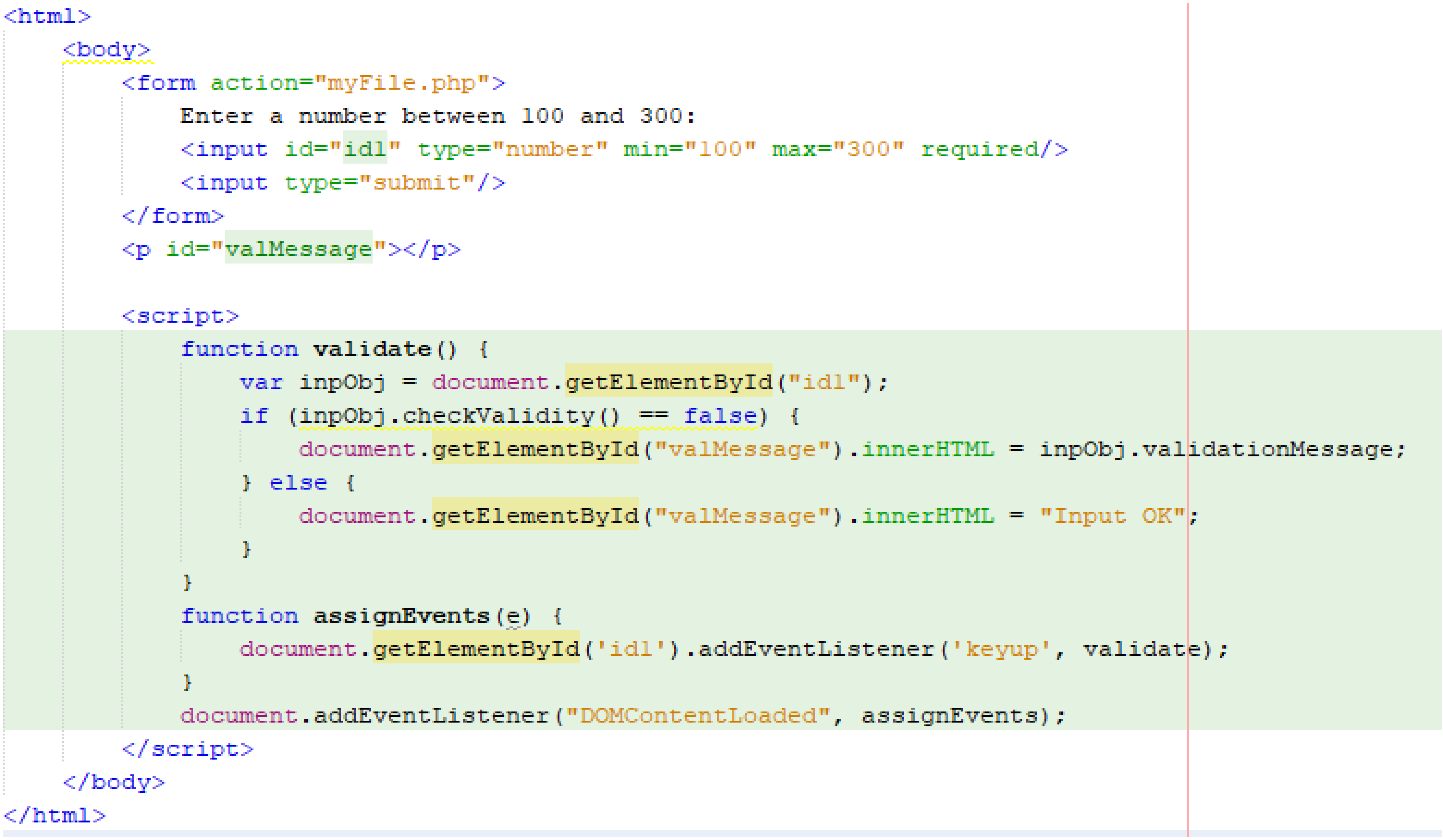


Attribute	Input types supporting the attribute
`pattern`	text, search, url, tel, email, password
`min`	range, number
	date, month, week
	datetime, datetime-local, time
`max`	range, number
	date, month, week
	datetime, datetime-local, time
`required`	text, search, url, tel, email, password, date, datetime, datetime-local, month, week, time, number, checkbox, radio, file; also on the `<select>` and `<textarea>` elements
`step`	date
	month
	week
	datetime, datetime-local, time
	range, number
`maxlength`	text, search, url, tel, email, password; also on the `<textarea>` element

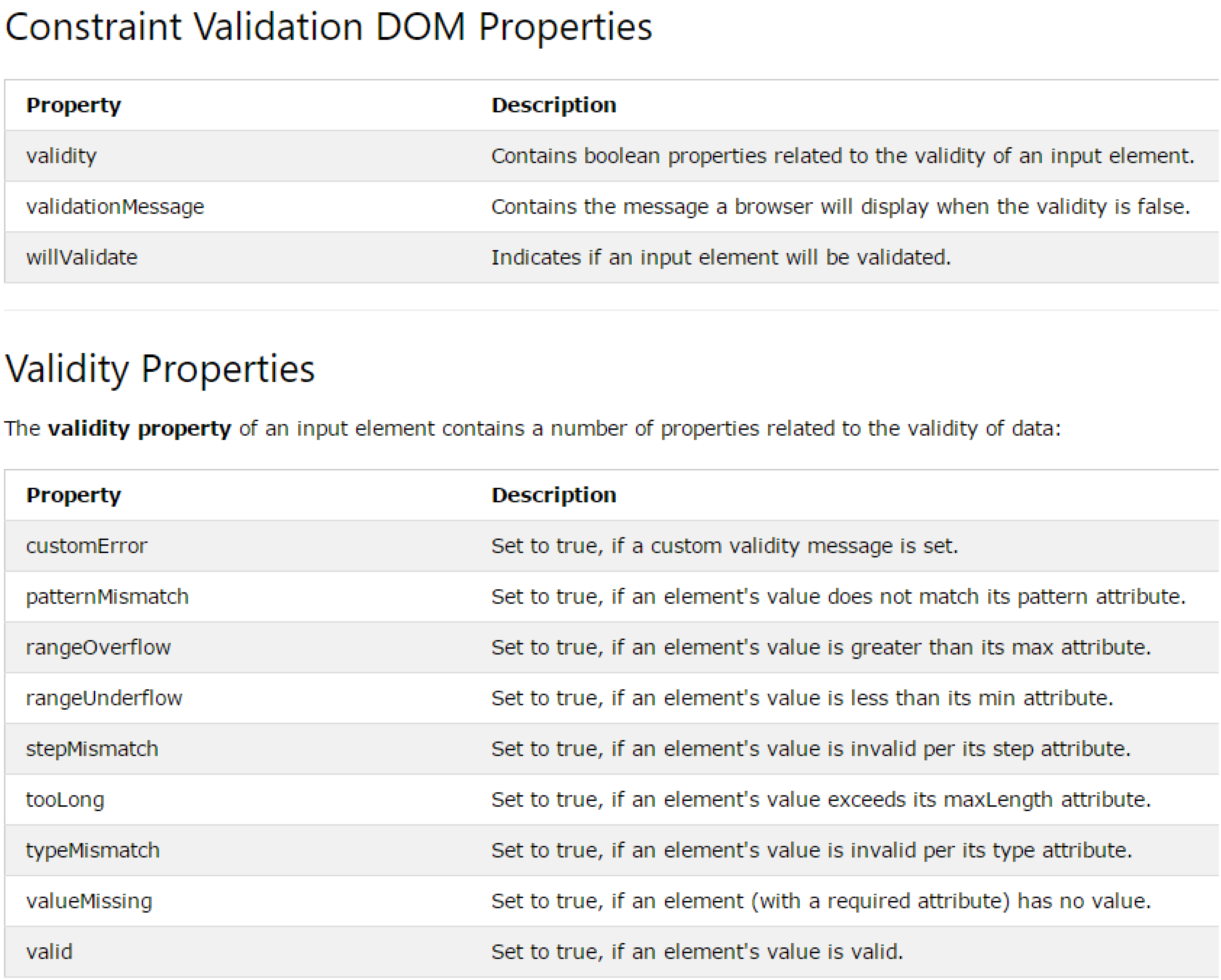
2. Constraint validation DOM methods: CheckValidity()

checkValidity()	Returns true if an input element contains valid data.

Example 1



```
<html>
    <body>
        <form action="myFile.php">
            Enter a number between 100 and 300:
            <input id="idl" type="number" min="100" max="300" required/>
            <input type="submit"/>
        </form>
        <p id="valMessage"></p>

        <script>
            function validate() {
                var inpObj = document.getElementById("idl");
                if (inpObj.checkValidity() == false) {
                    document.getElementById("valMessage").innerHTML = inpObj.validationMessage;
                } else {
                    document.getElementById("valMessage").innerHTML = "Input OK";
                }
            }
            function assignEvents(e) {
                document.getElementById('idl').addEventListener('keyup', validate);
            }
            document.addEventListener("DOMContentLoaded", assignEvents);
        </script>
    </body>
</html>
```



Constraint Validation DOM Properties

Property	Description
validity	Contains boolean properties related to the validity of an input element.
validationMessage	Contains the message a browser will display when the validity is false.
willValidate	Indicates if an input element will be validated.

Validity Properties

The **validity property** of an input element contains a number of properties related to the validity of data:

Property	Description
customError	Set to true, if a custom validity message is set.
patternMismatch	Set to true, if an element's value does not match its pattern attribute.
rangeOverflow	Set to true, if an element's value is greater than its max attribute.
rangeUnderflow	Set to true, if an element's value is less than its min attribute.
stepMismatch	Set to true, if an element's value is invalid per its step attribute.
tooLong	Set to true, if an element's value exceeds its maxLength attribute.
typeMismatch	Set to true, if an element's value is invalid per its type attribute.
valueMissing	Set to true, if an element (with a required attribute) has no value.
valid	Set to true, if an element's value is valid.

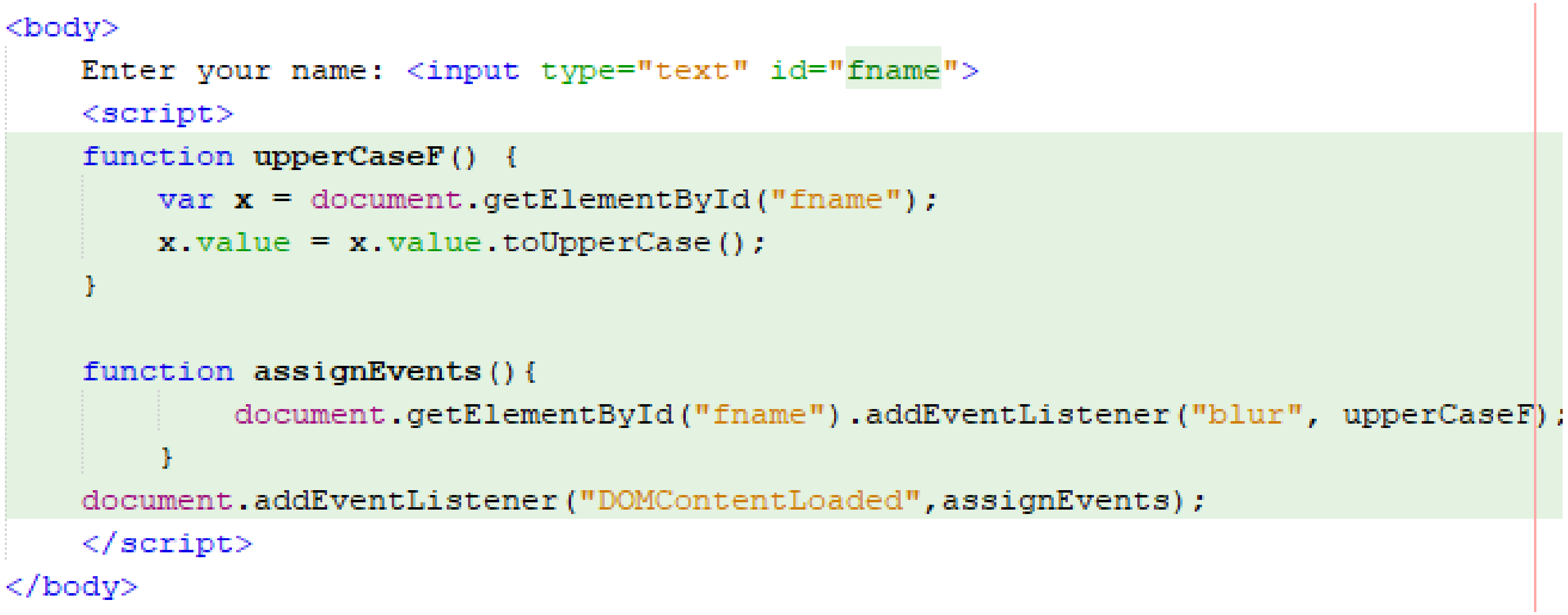
Javascript modification and validation

Coded using JavaScript: (https://www.w3schools.com/tags/ref_eventattributes.asp)

Attribute	Value	Description
blur	*script*	Fires the moment that the element loses focus
change	*script*	Fires the moment when the value of the element is changed
contextmenu	*script*	Script to be run when a context menu is triggered (right-click on an element to open the context menu)
focus	*script*	Fires the moment when the element gets focus
input	*script*	Script to be run when an element gets user input
invalid	*script*	Script to be run when an element is invalid
reset	*script*	Fires when the Reset button in a form is clicked
search	*script*	Fires when the user writes something in a search field (for <input="search">)
select	*script*	Fires after some text has been selected in an element
submit	*script*	Fires when a form is submitted

Form events

Example 2 (event *blur):* When the input field loses focus, a function is triggered which transforms the input text to upper case.

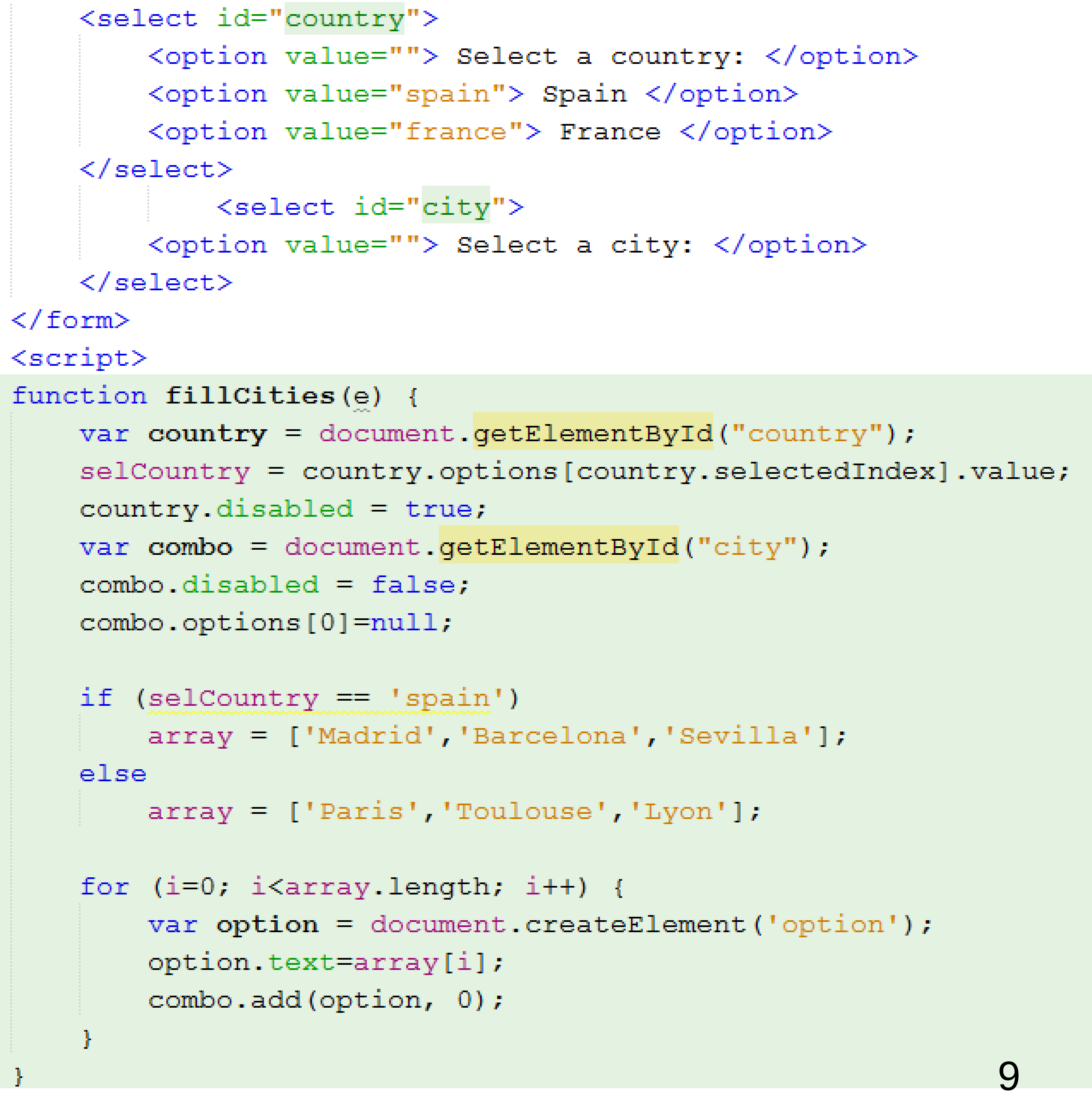


```
<body>
    Enter your name: <input type="text" id="fname">
    <script>
    function upperCaseF() {
        var x = document.getElementById("fname");
        x.value = x.value.toUpperCase();
    }

    function assignEvents(){
            document.getElementById("fname").addEventListener("blur", upperCaseF);
        }
    document.addEventListener("DOMContentLoaded",assignEvents);
    </script>
</body>
```

Form modification

Example 3 (Adding or removing options from a dropDown list):

Once the country is selected, the options are added to the city dropbox

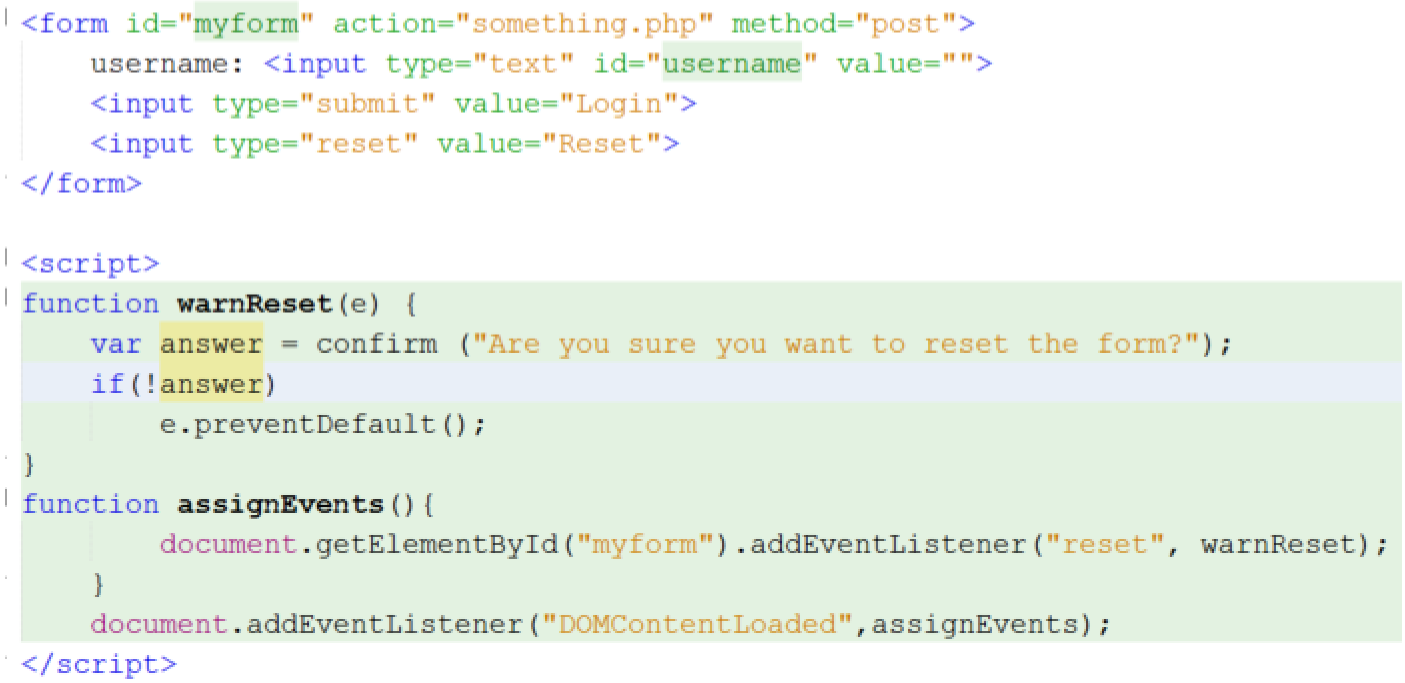


```
  <select id="country">
    <option value=""> Select a country: </option>
    <option value="spain"> Spain </option>
    <option value="france"> France </option>
  </select>
      <select id="city">
    <option value=""> Select a city: </option>
  </select>
</form>
<script>
function fillCities(e) {
  var country = document.getElementById("country");
  selCountry = country.options[country.selectedIndex].value;
  country.disabled = true;
  var combo = document.getElementById("city");
  combo.disabled = false;
  combo.options[0]=null;

  if (selCountry == 'spain')
    array = ['Madrid','Barcelona','Sevilla'];
  else
    array = ['Paris','Toulouse','Lyon'];

  for (i=0; i<array.length; i++) {
    var option = document.createElement('option');
    option.text=array[i];
    combo.add(option, 0);
  }
}
```

Form validation: PreventDefault method

The *preventDefault* method is used to require confirmation before submiting or reseting the form: If the user does not confirm, the action won’t be executed

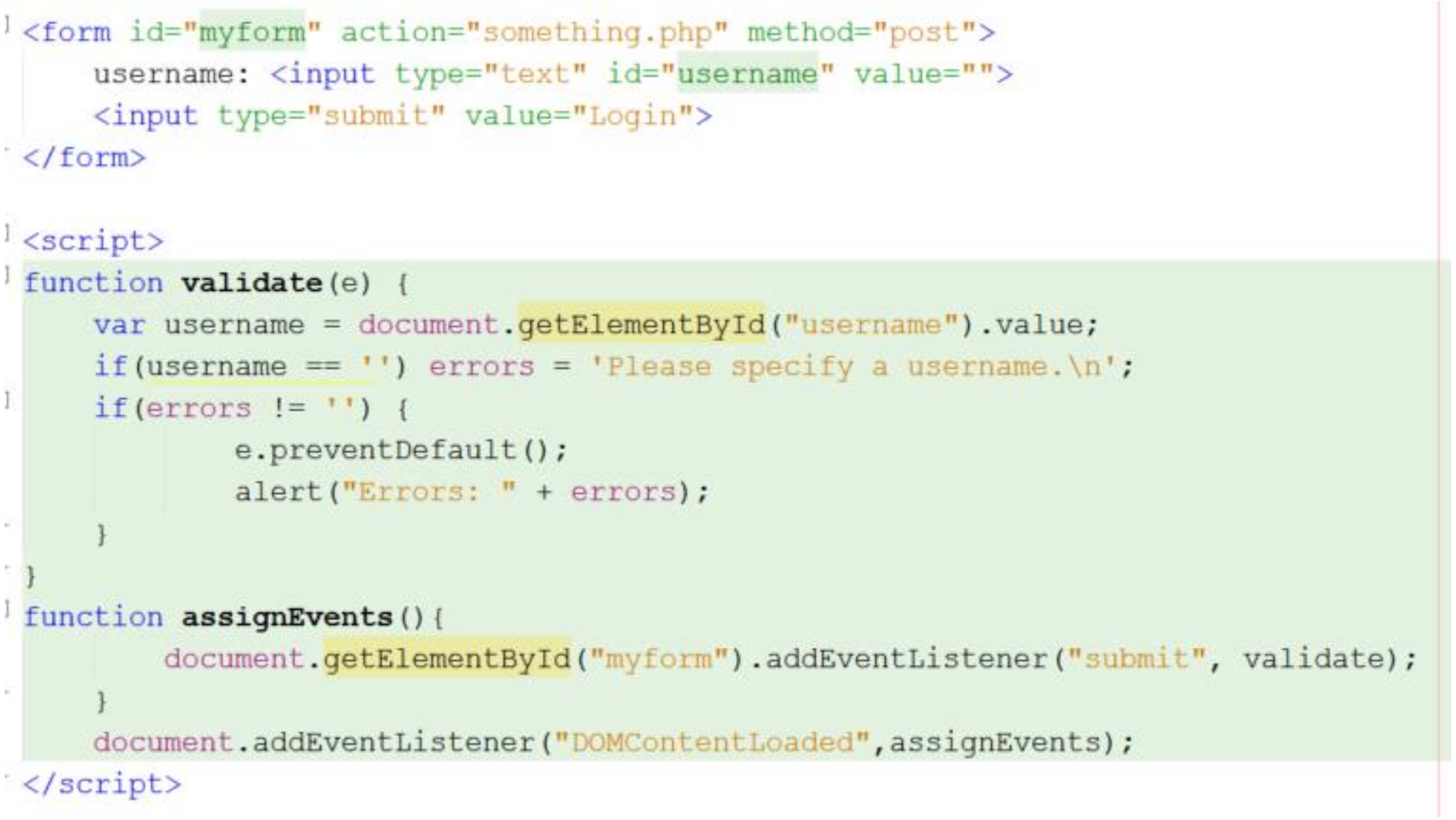
Example 4: Reset validation with *preventDefault* method



```
<form id="myform" action="something.php" method="post">
    username: <input type="text" id="username" value="">
    <input type="submit" value="Login">
    <input type="reset" value="Reset">
</form>

<script>
function warnReset(e) {
    var answer = confirm ("Are you sure you want to reset the form?");
    if(!answer)
        e.preventDefault();
}
function assignEvents(){
        document.getElementById("myform").addEventListener("reset", warnReset);
    }
    document.addEventListener("DOMContentLoaded",assignEvents);
</script>
```

Form validation

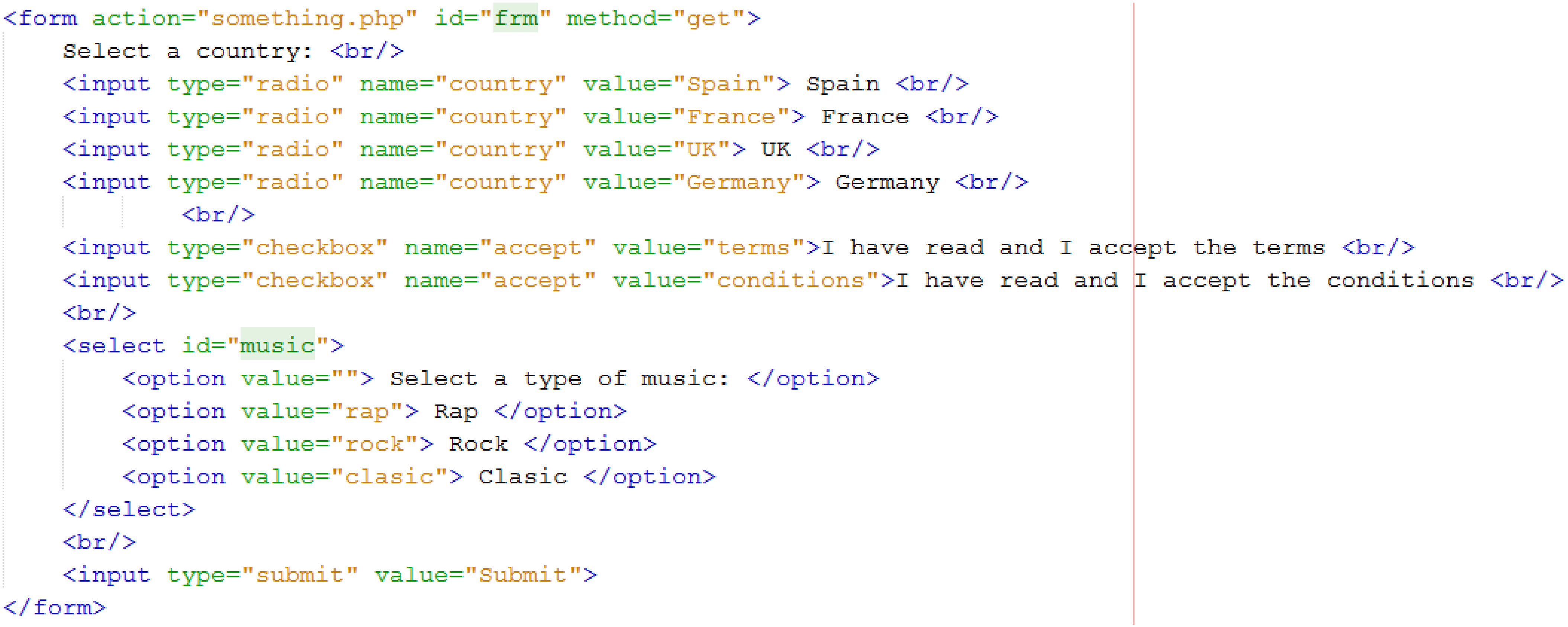
Example5: Validation before submission with *preventDefault* method:
If there are any errors, the form is not submitted, and a message will be shown
:



```
<form id="myform" action="something.php" method="post">
    username: <input type="text" id="username" value="">
    <input type="submit" value="Login">
</form>

<script>
function validate(e) {
    var username = document.getElementById("username").value;
    if(username == '') errors = 'Please specify a username.\n';
    if(errors != '') {
            e.preventDefault();
            alert("Errors: " + errors);
    }
}
function assignEvents(){
        document.getElementById("myform").addEventListener("submit", validate);
    }
    document.addEventListener("DOMContentLoaded",assignEvents);
</script>
```

Form validation

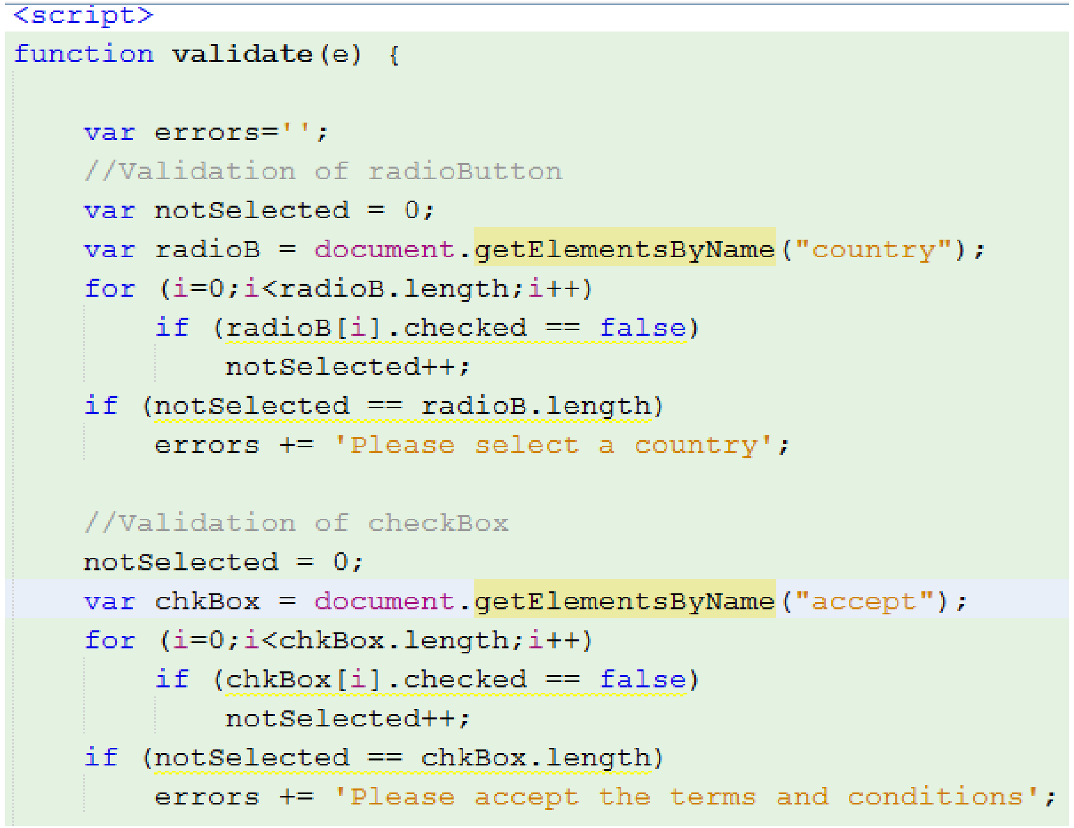
Example 6: Validation radioButton, checkBox and combo selections



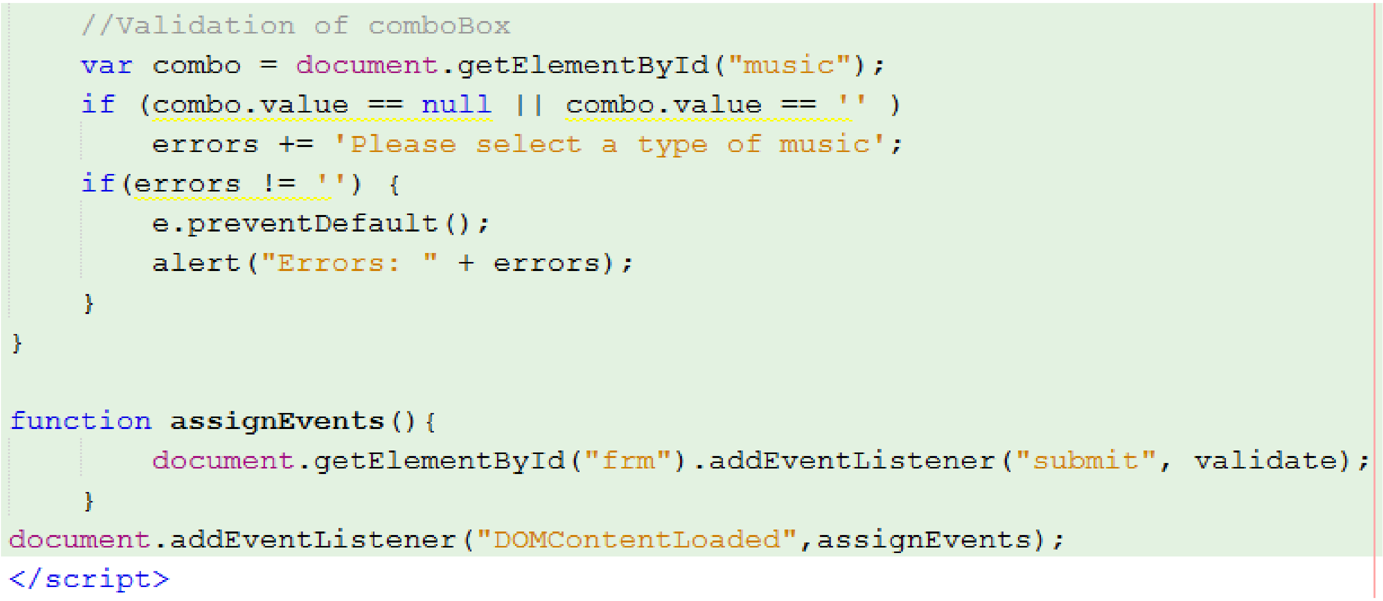
```
<form action="something.php" id="frm" method="get">
    Select a country: <br/>
    <input type="radio" name="country" value="Spain"> Spain <br/>
    <input type="radio" name="country" value="France"> France <br/>
    <input type="radio" name="country" value="UK"> UK <br/>
    <input type="radio" name="country" value="Germany"> Germany <br/>
            <br/>
    <input type="checkbox" name="accept" value="terms">I have read and I accept the terms <br/>
    <input type="checkbox" name="accept" value="conditions">I have read and I accept the conditions <br/>
    <br/>
    <select id="music">
        <option value=""> Select a type of music: </option>
        <option value="rap"> Rap </option>
        <option value="rock"> Rock </option>
        <option value="clasic"> Clasic </option>
    </select>
    <br/>
    <input type="submit" value="Submit">
</form>
```

Form validation

Validation radioButton, checkBox and combo selections (Cont):



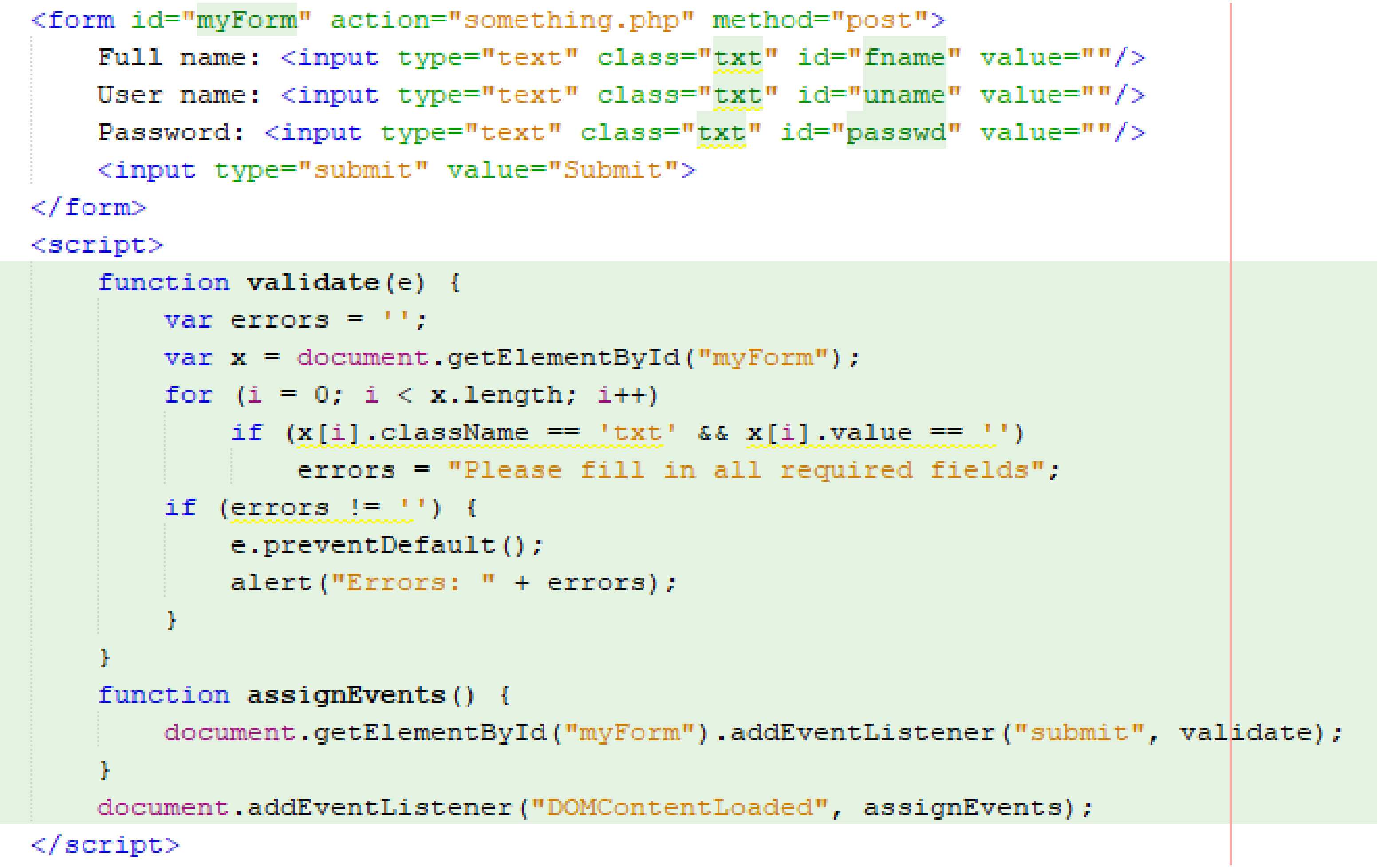
```
<script>
function validate(e) {

    var errors='';
    //Validation of radioButton
    var notSelected = 0;
    var radioB = document.getElementsByName("country");
    for (i=0;i<radioB.length;i++)
        if (radioB[i].checked == false)
            notSelected++;
    if (notSelected == radioB.length)
        errors += 'Please select a country';

    //Validation of checkBox
    notSelected = 0;
    var chkBox = document.getElementsByName("accept");
    for (i=0;i<chkBox.length;i++)
        if (chkBox[i].checked == false)
            notSelected++;
    if (notSelected == chkBox.length)
        errors += 'Please accept the terms and conditions';
```



```
    //Validation of comboBox
    var combo = document.getElementById("music");
    if (combo.value == null || combo.value == '' )
        errors += 'Please select a type of music';
    if(errors != '') {
        e.preventDefault();
        alert("Errors: " + errors);
    }
}

function assignEvents(){
        document.getElementById("frm").addEventListener("submit", validate);
    }
document.addEventListener("DOMContentLoaded",assignEvents);
</script>
```

Form validation

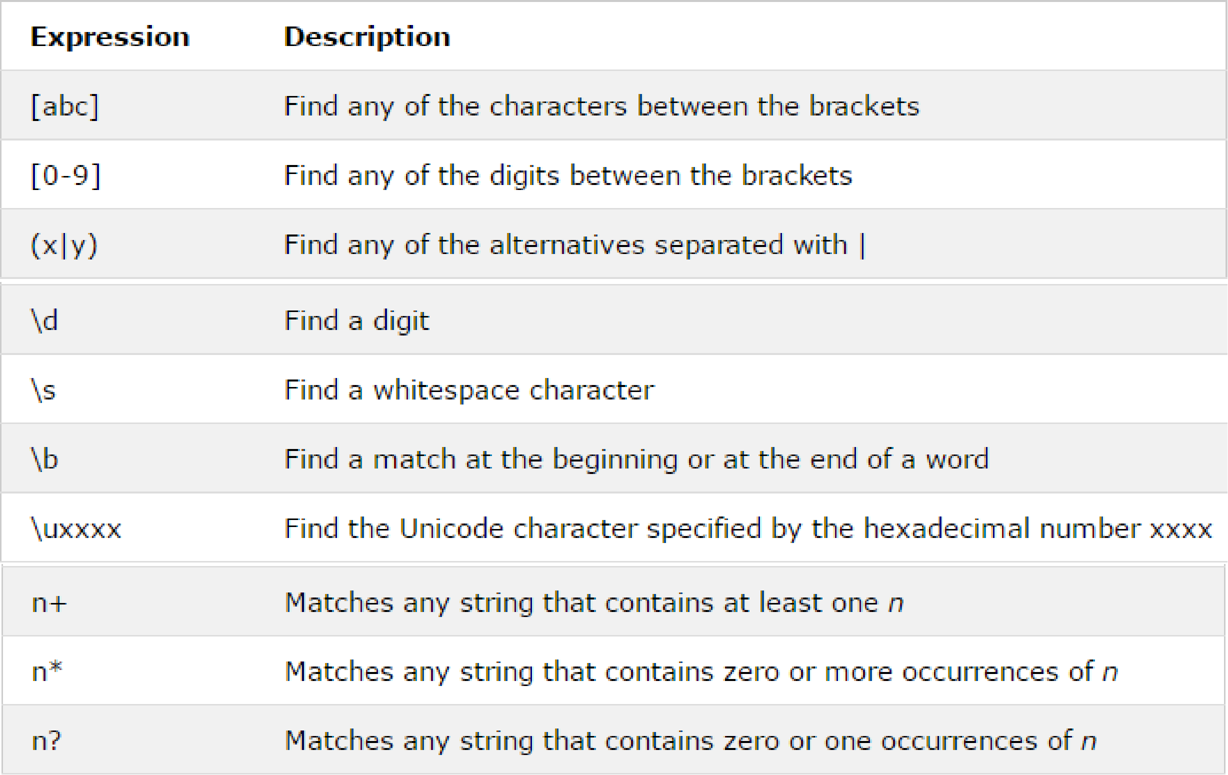
Example 7:
Form elements collection:
Checks if all text fields have been filled: If not, returns a message and does not submit the form



```
<form id="myForm" action="something.php" method="post">
    Full name: <input type="text" class="txt" id="fname" value=""/>
    User name: <input type="text" class="txt" id="uname" value=""/>
    Password: <input type="text" class="txt" id="passwd" value=""/>
    <input type="submit" value="Submit">
</form>
<script>
    function validate(e) {
        var errors = '';
        var x = document.getElementById("myForm");
        for (i = 0; i < x.length; i++)
            if (x[i].className == 'txt' && x[i].value == '')
                errors = "Please fill in all required fields";
        if (errors != '') {
            e.preventDefault();
            alert("Errors: " + errors);
        }
    }
    function assignEvents() {
        document.getElementById("myForm").addEventListener("submit", validate);
    }
    document.addEventListener("DOMContentLoaded", assignEvents);
</script>
```

Javascript regular expressions

A regular expression is a sequence of characters that forms a search pattern.
The search pattern can be used for text search and text replace operations, and for **validations**



Expression	Description
[abc]	Find any of the characters between the brackets
[0-9]	Find any of the digits between the brackets
(x|y)	Find any of the alternatives separated with |
\d	Find a digit
\s	Find a whitespace character
\b	Find a match at the beginning or at the end of a word
\uxxxx	Find the Unicode character specified by the hexadecimal number xxxx
n+	Matches any string that contains at least one *n*
n*	Matches any string that contains zero or more occurrences of *n*
n?	Matches any string that contains zero or one occurrences of *n*

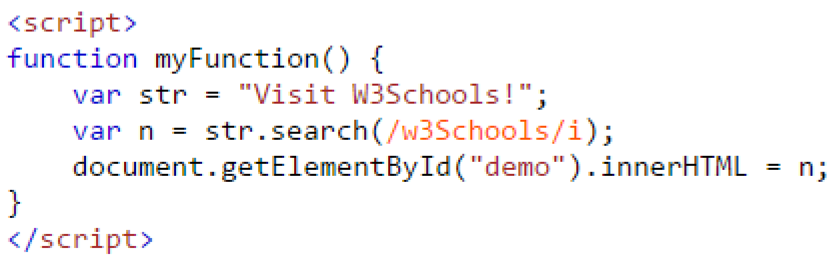
Complete reference: https://www.w3schools.com/jsref/jsref_obj_regexp.asp

Javascript regular expressions

- Syntax:

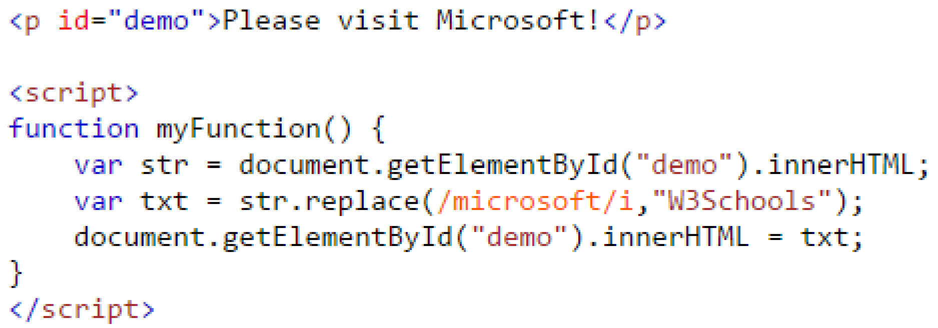
  /pattern/modifiers;

- **The search() method** uses an expression to search for a match, and returns the position of the match.



```
<script>
function myFunction() {
    var str = "Visit W3Schools!";
    var n = str.search(/w3Schools/i);
    document.getElementById("demo").innerHTML = n;
}
</script>
```

- **The replace() method** returns a modified string where the pattern is replaced.

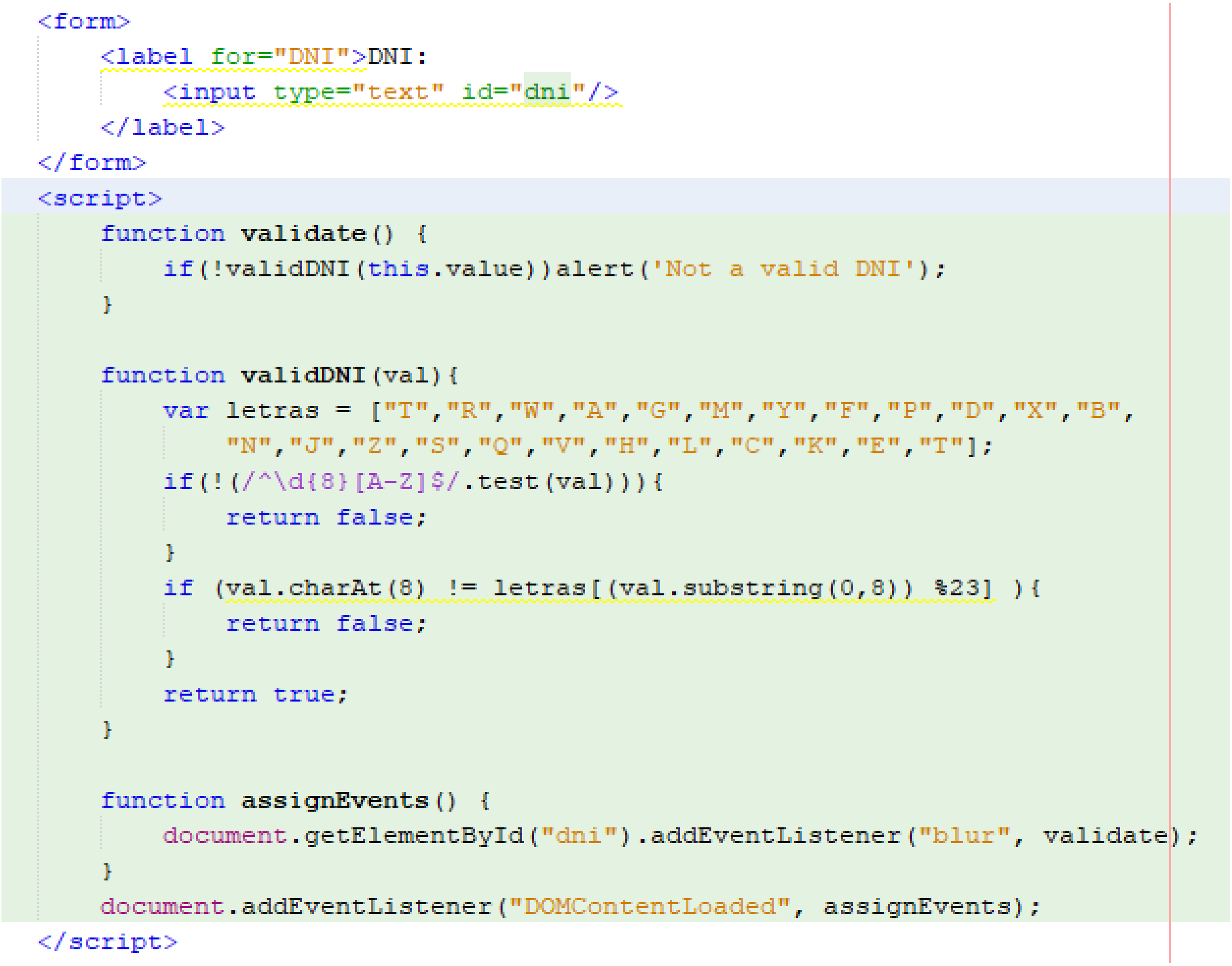


```
<p id="demo">Please visit Microsoft!</p>

<script>
function myFunction() {
    var str = document.getElementById("demo").innerHTML;
    var txt = str.replace(/microsoft/i,"W3Schools");
    document.getElementById("demo").innerHTML = txt;
}
</script>
```

Validating against a regular expression

The test() method uses an expression to validate a format, and returns true or false

Example_8 : Validates DNI format using the ***test*** method



```
<form>
    <label for="DNI">DNI:
        <input type="text" id="dni"/>
    </label>
</form>
<script>
    function validate() {
        if(!validDNI(this.value))alert('Not a valid DNI');
    }

    function validDNI(val){
        var letras = ["T","R","W","A","G","M","Y","F","P","D","X","B",
            "N","J","Z","S","Q","V","H","L","C","K","E","T"];
        if(!(/^\d{8}[A-Z]$/.test(val))){
            return false;
        }
        if (val.charAt(8) != letras[(val.substring(0,8)) %23] ){
            return false;
        }
        return true;
    }

    function assignEvents() {
        document.getElementById("dni").addEventListener("blur", validate);
    }
    document.addEventListener("DOMContentLoaded", assignEvents);
</script>
```