Front-End Development
Exercise 06
HTTP and Forms

Help to complete the tasks of this exercise can be found on the chapter 18 ” HTTP and Forms” of our course book “Eloquent JavaScript” (3rd edition) by Marijin Haverbeke. The aims of the exercise are to understand HTTP and Form in JavaScript-context.

Embed your theory answers, drawings, codes and screenshots directly into this document. Always immediately after the relevant question. Return the document into your return box in the Optima platform by the deadline.

It’s also recommendable to use Internet sources to supplement the information provided by the course book.

The maximum number of points you can earn from this exercise is 10.

Tasks:

1. Answer to following questions (2 pts):
    a. List and compare different HTTP *methods*?

PUT	Replaces all representations of the target resource with the request payload
HEAD	Asks for response, same as get, but without response body
DELETE	Deletes specific resource
CONNECT	Establishes tunnel to the server identified by target resource
OPTIONS	Describes communication options for the target resource
TRACE	Performs a message loop-back test along the path to the target resource
PATCH	Applies partial modification to the resource.

    b. When the Get and Post methods should be used

    You use GET to ….. get something, without modifying it. You view something. POST is when you want to modify.
    You could use POST to update for example information about your account.
    Where you would use GET for example when you use search engine.

    c. You have seen the HTTP response code 404. What it means and what are the possible reasons for that?
    It is HTTP's standard response code that means browser was able to locate server but was unable to load what was requested. It usually comes with "page not found".
    It can also be in scenario where you don't to tell if the website exists or not if the person who tries to access it does not have rights to access it.

    d. How the values filled in the form by the user can be send over the HTTP?
    You can send it by using POST method, which sends updated data

2. Alternative a: Create a form for data collection. Form should have inputs for: (2 pts)
    a. User contact information✓
    b. 5 questions where user chooses numeric value between 1-3 using radio input (hint: grouping) ✓
    c. Text area for free feedback ✓

```
<!DOCTYPE html>
<html>
<head>
    <link rel="sylesheet" href="EX06T6.css">
</head>
<body>
    <form onsubmit="submitForm(event)">
        <fieldset>
            <legend>Required information</legend>
            <p>
                <legend for="test">What is your first name?</legend>
                <input type="text"  id="test" reguired name="first name"
pattern="[A-Z][a-z]{1,15}" required>
            </p>
            <p>
                <legend for="last name">What is your last name?</abbr></legend>
                <input type="text" id="last name" reguired name="last name"
pattern="[A-Z][a-z]{1,15}"required>
            </p>
            <p>
                <legend for="age">How old are you?</legend>
                <input type="number"  min="7" max="120" step="1" id="a1" reguired
name="age" pattern="\d+"required>
            </p>
            <p>
                <legend for="email">What's your e-mail address?</legend>
                <input type="email" id="email" name="email">
            </p>

            <p>
                <legend>Do you like coding?<abbr title="This field is mandatory"
aria-label="required"></abbr></legend>
                <input type="radio" required name="code" id="r1" value="Yes"><label
for="r1">Yes</label>
                <input type="radio" required name="code" id="r2" value="No"><label
for="r2">No</label>
                <input type="radio" required name="code" id="r3"
value="Maybe"><label for="r3">Maybe</label>
            </p>
            <p>
```

```
                <legend>Do you exercise often?<abbr title="This field is mandatory"
aria-label="required"></abbr></legend>
                <input type="radio" required name="exe" id="r1" value="Yes"><label
for="r1">Yes</label>
                <input type="radio" required name="exe" id="r2" value="No"><label
for="r2">No</label>
                <input type="radio" required name="exe" id="r3"
value="Sometimes"><label for="r3">Sometimes</label>
            </p>
            <p>
                <legend>Do you play videogames?<abbr title="This field is
mandatory" aria-label="required"></abbr></legend>
                <input type="radio" required name="games" id="r1"
value="yes"><label for="r1">Yes</label>
                <input type="radio" required name="games" id="r2" value="no"><label
for="r2">No</label>
                <input type="radio" required name="games" id="r3"
value="Sometimes"><label for="r3">Sometimes</label>
            </p>
            <p>
                <legend>Have you enjoyed your studies?<abbr title="This field is
mandatory" aria-label="required"></abbr></legend>
                <input type="radio" required name="study" id="r1"
value="Yes"><label for="r1">Yes</label>
                <input type="radio" required name="study" id="r2" value="No"><label
for="r2">No</label>
                <input type="radio" required name="study" id="r3" value="It has ups
and downs"><label for="r3">It has ups and downs</label>
            </p>
            <p>
                <legend>Do you feel motivated?<abbr title="This field is mandatory"
aria-label="required"></abbr></legend>
                <input type="radio" required name="driver" id="r1"
value="Yes"><label for="r1">Yes</label>
                <input type="radio" required name="driver" id="r2"
value="No"><label for="r2">No</label>
                <input type="radio" required name="driver" id="r3"
value="Sometimes"><label for="r3">Sometimes</label>
            </p>
        </fieldset>
        <p>
          <legend for="feed">You can leave feedback!</legend>
          <textarea id="feed" name="msg" maxlength="140" rows="5"></textarea>
        </p>
        <p>
          <button>Submit</button>
        </p>
    </form>
    <script type="text/JavaScript">
```

```
        function submitForm(event){
          event.preventDefault();
          window.alert("Hähä, eipäs paineta ;)");
          //window.history.back();
        }
    </script>
</body>
</html>
```

3. Explain how you prevent a form submission using JavaScript and why would you do that? Give also a simple code example (1 pts).

   You might want to prevent form submission if it contains illegal characters, might include javascript in the form or the form is empty.

   You can prevent sending form for example using eventlistener with preventDefault

```
<form onsubmit="submitForm(event)">
    <input type="text">
    <button type="submit">Submit</button>
  </form>
  <script type="text/JavaScript">
    function submitForm(event){
      event.preventDefault();
      window.history.back();
    }
  </script>

element.addEventListener("submit", function(event) {
    event.preventDefault();
    window.history.back();
  }, true);
```

4. What means local storage related to JavaScript? How could you use it? Write an example code that (2 pts)

   You can use local storage to store information to session or to browser. It helps you to save data, if something unexpected happens, such as browser crashes or computer crashes. You can choose between browser saving and session saving.

   a. has one input for text
   b. on blur-event writes contents in the local storage
   c. another code reads it when window is loaded
   d. Demonstrate by refreshing window…

```html
<!DOCTYPE html>
<html>
<head>
    <link rel="sylesheet" href="EX06T6.css">
</head>
<body>
    <form>
        <p>
            Enter your feedback: <input type="text" id="feedback" onblur="mysave()">
        </p>
    </form>
    <script>
        myload();
        function mysave() {
            alert("Saved")
          var x = document.getElementById("feedback").value;
          localStorage.setItem("MyFeedback", x);
        }
        function myload() {
            if (localStorage) {
            alert("Loadeds")
            var x = localStorage.getItem("MyFeedback");
            document.getElementById("feedback").value=x;
            }


        }
        </script>
</body>
</html>
```

5. Form Validation (1 pts)
    a. What is form Validation. (0.5 pts)

        It checks if the specific form “slot” is correctly filled, for example it can use RegEx to make sure there is correct characters. It also helps to keep database “clean”.

    b. Why we should use it (0.5pts)

        It keeps database clean, and we will get right information. There is also cases where people have highjacked sites using forms, that is also why you need to validate, so the possible input does not corrupt your system.

6. Check the link below and do the following (2 pts)

   https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation

   a. On the page section **Validate forms using JavaScript,** use the code given in "A more detailed example", and modify it. So, the email input field has to validate the maxlength of 50 characters. (1pts)

   b. On the page section **Validating forms without a built-in API**, use the code there and also apply the maxlength to the email address. (1pts)

**^(?=.{0,50}$)[**a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$

```
<body>
  <form novalidate="">
    <p>
      <label for="mail">
        <span>Please enter an email address:</span>
        <input type="email" id="mail" name="mail" required="" minlength="9" maxlength="20">
        <span class="error" aria-live="polite"></span>
      </label>
    </p>
    <button>Submit</button>
  </form>
```

Code required for Q2

```
<!DOCTYPE html>
<html lang="en">
 <head>
<title>Search a car</title>
<meta charset="UTF-8">
</head>
<body>
<h2>Search a car</h2>
<form id="carform" action="http://www.donotexists.fi/donotexists" novalidate>
<p><label for="mailaddr">Email:
<input id="mailaddr" name="mailaddr" type="email" size="30" required /> </label></p>
<p><label for="carcolor">Car color:
<select id="carcolor" name="carcolor">
<option value="" selected="selected">Select a color</option>
<option value="b">Blue</option>
<option value="r">Red</option>
</select>
</label></p>
 <p>
<label for="sunroof">
<input type="checkbox" id="sunroof" name="sunroof" value="yes" />Sunroof
</label>
<label for="ewindows"><input type="checkbox" id="ewindows" name="sunroof" value="yes" />Electronic windows</label> </p>
<p>
<label for="doors">Number of doors&amp;nbsp;&amp;nbsp;
<input type="radio" name="doors" id="door2" value="door2" class="radio" />Two
<input type="radio" name="doors" id="door4" value="door4" class="radio" />Four
</label></p>
 <p><label for="zipno">Zip number:
<input id="zipno" name="zipno" type="text" size="5" maxlength="5" /><br/><br />
</label></p>
<input type="submit" value="Send" /> &amp;nbsp;<input type="reset" value="Clear" />
```

```
<p id="validationinfo"></p>

</form>

</body>

</html>
```