(/)

We're moving to Discord!

In a few days, we will be leaving Slack in favor of Discord 🎉

👆 **Click here for more information (/concepts/100033)**

HTML Form + JS validation

HTML **JavaScript**

Master

By: Dr. Ehoneah Obed

Weight: 1

Project will start Sep 22, 2023 10:00 PM, must end by Sep 28, 2023 9:59 PM

☑ **Manual QA review must be done** (request it when you are done with the project)

Resources

Read or watch:

- Javascript Form Validation (/rltoken/_iQ-AFZJcZTyaY3Ad4aOQA)
- Javascript Forms (/rltoken/z_W5ZiXIB9VEu4GnBKY_Bw)
- How to make form validation using Javascript - video (/rltoken/FkaOrbNNr0_D-JPbK5hPgA)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/amr8JGu72kcxqcvArkpb3A), **without the help of Google**:

General

- What client-side validation is
- How to implement basic form validation using JavaScript.
- How to handle form submissions and perform data validation using JavaScript.

Requirements

General

- Recommended editors: `Visual studio code`
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project is mandatory
- You are **not allowed** to install, import or use external libraries. This website must be build with only HTML/CSS/JavaScript. No NodeJS, React, VueJS, Bootstrap, etc.
- Your code should be W3C compliant and validate with W3C-Validator (/rltoken/5lnDHcfQ9_XKwXPv0rjB2w)

Tasks

0. Form Validation - Password Strength

mandatory

Below is the content of `password.html` :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form id="passwordForm">
  <label for="password">Password:</label>
  <input type="password" id="password" name="password">
  <button type="submit">Submit</button>
  <p id="error" style="color: red;"></p>
</form>
</body>
</html>
```

- Implement JavaScript validation to check the password strength when the form is submitted.
- Retrieve the value entered in the password input field using JavaScript.
- Your JavaScript should be written in a separate file and used as an external script in the HTML code
- Write a function called validatePassword to validate the password based on the provided criteria.
- The validatePassword function should check the following criteria:
- The password must be `at least 8 characters long`.
- The password must contain `at least one uppercase letter`.
- The password must contain `at least one lowercase letter`.

- The password must contain `at least one numeric digit`.
- The password must contain `at least one special character (e.g., !@#$%^&*)`.
- If the password meets all the criteria, allow the form submission.
- If the password does not meet the criteria, display an appropriate error message in the `<p>` element with the id “error” .

Repo:

- GitHub repository: `alx_javascript`
- Directory: `html_js_validation`
- File: `0-password-strength.js, password.html`

Help

0/14 pts

1. Form Validation - Email Validation

mandatory

Below is the content of `email.html` :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form id="emailForm">
  <label for="email">Email:</label>
  <input type="text" id="email" name="email">
  <button type="submit">Submit</button>
  <p id="error" style="color: red;"></p>
</form>
</body>
</html>
```

- Implement JavaScript validation to check the email format when the form is submitted.
- Retrieve the value entered in the email input field using JavaScript.
- Your JavaScript should be written in a separate file and used as an external script in the HTML code
- Write a function called `validateEmail` to validate the email based on the email format.
- The `validateEmail` function should check if the email follows the standard email format, which typically includes an alphanumeric `username` , followed by the `@` symbol, and then the `domain name` .
- If the email format is `correct` , allow the form submission.

- If the email format is `incorrect`, display the error message “ `Please enter a valid email address.` ” in the element with the id `"error"`.
- Prevent the default form submission behavior using JavaScript to prevent the form from submitting if the email format is not valid.

Repo:

- GitHub repository: `alx_javascript`
- Directory: `html_js_validation`
- File: `1-email-validation.js, email.html`

Help **0/13** pts

2. Dynamic Form Field

mandatory

Below is the content of `dynamic-form.html` :

```
<form id="dynamicForm">
  <label for="numFields">Number of Fields:</label>
  <select id="numFields" name="numFields">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
  </select>
  <div id="inputContainer"></div>
  <button type="submit">Submit</button>
</form>
```

Implement JavaScript to generate a dynamic number of text input fields based on the selected value in the dropdown menu.

- Retrieve the selected value from the dropdown menu using JavaScript.
- Write a function called `generateInputFields` to generate the specified number of text input fields dynamically within the “inputContainer” div.
- The `generateInputFields` function should take the selected value from the dropdown menu as a parameter.
- Inside the `generateInputFields` function, use a loop or other appropriate logic to create the desired number of text input fields.
- Each text input field should have a unique name attribute, such as “field1”, “field2”, etc., to differentiate them.
- Append the dynamically generated text input fields to the “inputContainer” div.
- Implement JavaScript validation to ensure that all dynamically generated fields are filled before submitting the form.

- Write a function called `validateForm` to validate the form before submission.

- The `validateForm` function should be called when the form is submitted.
- Inside the validateForm function, retrieve the values entered in the dynamically generated text input fields using JavaScript.
- Check if any of the fields are empty. If any field is empty, prevent the form submission and display the error message “ `Please fill in all fields` ”.

Note: You may need to use event listeners or other appropriate techniques to detect changes in the dropdown menu selection and handle form submission.

Repo:

- GitHub repository: `alx_javascript`
- Directory: `html_js_validation`
- File: `2-dynamic-form.js, dynamic-form.html`

Help **0/20** pts

3. Form Submission Handling

mandatory

Below is the content of `form-submission.html` :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form id="submitForm">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>
  <!-- Add more form fields as needed -->
  <button type="submit">Submit</button>
</form>
</body>
</html>
```

Implement JavaScript to handle form submission and perform validation before submitting.

- Write a function called `handleFormSubmit` to handle the form submission.
- Your JavaScript should be written in a separate file and used as an external script in the HTML code.
- Add an event listener to the form element to listen for the “submit” event and call the `handleFormSubmit` function.

(/)

- Inside the `handleFormSubmit` function, prevent the default form submission behavior using `event.preventDefault()`.
- Retrieve the values entered in the form fields using JavaScript.
- Use JavaScript validation to ensure that all required fields are filled.
- Check if the "name" and "email" fields are filled. If any of them are empty, display the error message "`Please fill in all required fields`".
- If all validations pass, display a success message "`Form submitted successfully!`".
- If any validations fail, do not submit the form and allow the user to correct the errors before submitting again.

Note: You may need to use the DOM manipulation methods, such as getElementById(), to retrieve form field values and update the DOM to display error messages or success messages.

Repo:

- GitHub repository: `alx_javascript`
- Directory: `html_js_validation`
- File: `3-form-submission.js, form-submission.html`

Help

0/14 pts

Score

Now that you are ready to be reviewed, share your link to your peers. You can find some here.

https://intranet.alxswe.com/corrections/16988927/correct

Don't forget to review one of them (/corrections/to_review).

The next project will be available on Saturday, Sep 30th.

Skip this project

Previous project (/projects/100064)