

BUILD A SIMPLE WEBSITE

Your task is to implement the simple webpage according to functional requirements and your best knowledge.

Source code must be located in the github repository. If you don't have a github account, create one at <https://github.com/>.

Write your solution using web core HTML, CSS and JavaScript. You have 4 days to deliver a solution. When the task is finished please send a github link as a reply.

There are some requirements you should fulfill:

- The page should contain at least 4 subpages.
 - Its html should be W3C valid.
 - It should contain properly used HTML5 elements
 - It should contain a navigation menu
 - It should contain a contact page with a form to send email to the author (**you don't need to create server-side logic, only the front end**)
 - Optional: The contact page should have validation written in JavaScript using regular expressions
 - Optional: It should contain a welcome screening which displays an animation
-
- **DO NOT use any CSS frameworks (e.g. Bootstrap)**
 - **DO NOT use any JavaScript frameworks (e.g. jQuery, React)**
-
- Remember to use sources (images, sounds etc.) only with proper usage rights, labeled 'for reuse', and give attribution to the original author. You can find many websites with free photos and graphics to commercial use, e.g. pixabay

Step by step:

Create your index page

1. Before you start coding, sketch up how your web pages should look like. You can do it on paper or with any wireframe application. This is your choice. The purpose is to visualize different sections and parts that will be on your page and how should they be arranged.
2. Create a new html file in your repository root directory named: index.html
3. Set the doctype to HTML5
4. Add `<head>` part and setup basic metadata and the `<title>` of the page
5. Add `<body>` part, which will include below points.
6. Add `<header>` to the body with the navigation menu. The first menu item should link to the index page.
7. Add a `<div>` element to the body which has a class named: **content**. This will contain the part of the pages which differ from each other.
8. Add an image of you and a small introduction next to it about yourself.
9. Add some content to it (e.g. the description of the exercise) to see where it will be. You can fill it with arbitrary content. Take care to use the different html tags properly so it can express your intent.
10. Add `<footer>` and add some content to it. Take an example from other sites.
11. After you finish commit your changes into your repository with a descriptive comment.

Make it nice

1. Create a new folder named CSS in your repository's root directory
2. Create a **style.css** file in that newly created folder. This will contain all your rules about your site's appearance.
3. Add a reference to it in your **index.html** file's head part.
4. Define basic rules for the **body** element. (Default background, font, alignment...) These rules are inherited in the child elements (every element located inside the body).
5. Write CSS rules to make all the sites like your sketch.

Create a contact page

1. Copy your index.html file and give it the name: contact.html
2. Add a link to your navigation menu which redirects to this html page
3. Delete the inner content of **div** with class **content** element
4. Create a form inside it
5. Add two `<input>` fields and a `<textarea>` field to the form and proper `<label>`s to them. The first input field is the name which has a placeholder text: *Enter your name* The second input field is the email with placeholder text: *Give your email* The textarea will contain the message so give some handful instruction in the placeholder text. Do not forget to add names to all your form elements.

6. Add a submit button which should have the text SEND on it. It should be disabled by default.
7. Open your **style.css** file and at the bottom of it and write some rules which makes it nice. (e.g. horizontally centered and the fields have the same width). You can find references and inspiration on [Behance](#) or Dribble.
8. Commit your changes to the repository.

OPTIONAL - Add JavaScript form validation to contact page

1. Write code that will enable the submit button if all form fields are not empty. It should get back to being disabled if the content of any field will be empty again.
2. Write JavaScript code using regular expressions that will check if name fulfills following restrictions:
 - it is not empty and does not contain only whitespace characters
 - it contains only letters and space
 - each word in it starts with capital letters
3. Write code using regular expressions that will check if email fulfills following restrictions:
 - it is not empty and does not contain only whitespace characters
 - it is a valid email address
4. Write code that will fire validation methods for each field when the submit button is pressed. It should not cause page reloading.
5. If there are validation errors after clicking SEND, display them on the form or popup.
6. If there are no errors after clicking SEND, the page should display an alert with information that the message has been sent. After that, each field of form should be cleared.
7. After you finish commit your changes into your repository with a descriptive comment. If you feel that it would be good to have separate commits for each step, do it.

Create your custom pages

1. Create your new HTML file(s) based on the index.html
2. Add the proper link to the navigation menu on every other html files.
3. Fill it with custom content
4. Extend your stylesheet if needed
5. Do not forget to commit and push your changes into your repository!

Validate your pages

1. Go to <https://validator.w3.org/>
2. Validate your sites
3. Fix the issues

While checking your implementation, we will give you feedback on conforming requirements and on the UX/UI of your solution.

Hosting your page

If you want, you can try out hosting your page online using [GitHub Pages](#). It's used by many developers to host different types of static pages. You can find many homepages of open source projects hosted as a page on GitHub Pages.

Good luck!

Motorola Academy team



MOTOROLA
SOLUTIONS

