Programming in HTML5 with JavaScript and CSS3

Challenges STEP BY STEP

Module 1: HTML, DIVS & Background-color

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the html, head and body tags into a new html document.

Task 2: Display a face of a Rubik cube

Step 1

Go to your solutions Folder and copy the Rubik's face image in the same folder than your html document.

Step 2

Display the image on your browser by using the img tag and the src property pointing to your image.

Task 3: Build the Rubik's face using only html elements

Step 1

Use div or span elements and some CSS styling to create a grid of 3x3.

Step 2

Color each area of the elements corresponding to the Rubik's cube face image with the background color property.

Task 4: Play Tic Tac Toe

Step 1

Fill in the contents of the elements with an X or an O.

Step 2

Center the X and the O letters in all the elements and color them differently.

Module 2: Introduction to position property

Lab: Positioning HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the html, head and body tags into a new html document.

Task 2: Display the sheeps, wolfs and wall images

Step 1

Go to your solutions Folder and copy the sheeps, wolfs and wall images in the same folder than your html document.

Step 2

Display the images on your browser by using the img tag and the src property pointing to your image rendering at least 3 wolfs, 6 sheeps and one wall.

Task 3 Class or identify your images

Step 1

By using the class or id properties, create a way to target your images with CSS.

Task 4 Position the elements on screen

Step 1

By using CSS styles, position the wolfs and the sheeps at opposite sides of the browser viewport

Step 2

By using CSS styles, position the wall between the sheeps and the wolfs.

Module 3: HTML5 tags

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2: Creative process

Step 1

Read carefully the suggestions for a project on the instructions, you can design a blog, a social network or a streaming platform.

Step 2

Depending on your choice, do some research online and compare other blogs, social networks or streaming platforms and identify the key components of those websites. For example the position and size of thubmbails or video players.

Task 3: Mock up your website

Step 1

By using any tool mockup the sections of your website similar to the example on the lab instructions. Some suggestions include Photoshop or GIMP.

Step 2

Build that structure using your HTML5 elements.

Step 3

Apply styles to your elements so they match with your mockup.

Bonus Step

Try to add some content to your mockup, such as sample text and random images or videos.

Module 4: Forms

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Define user data

Step 1

Choose from 2 to 3 user data you consider your web project need to register your users.

Task 3 Create the form

Step 1

Create your form with the form and input tags and use the type attribute of the input tags to specify what kind of data you're capturing. Don't forget to add an input type submit at the end.

Step 2

Apply CSS styles to make your form look original:

- Change the background color of the input field.
- Change the size of the input field.
- Add some roundness to the borders of your input fields.

Step 3

Test filling your form on your browser with different no valid values and click on submit to verify the default behavior of your form.

Module 5: Advanced selectors in css

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Add media to your project

Step 1

Copy the logo, the clock and the hand images from the Solutions folder to your project folder.

Step 2

Insert an img element and render each one of the resources.

Step 3

Apply some CSS styles so the hand is centered in the middle of the clock. You can also use other approach, for example setting the clock image as a background-image of a div.

Step 4

Create a table with any content in a 3x6 grid by using the table, thead, tbody, tr, th and td elements. You can also simplify it by using only the table, tr and td elements.

Task 3 Animate and style

Step 1

Apply CSS styles to the logo to animate it. You can do it by using the hover selector on the styles of the logo. It’s a bonus if you use the animation CSS property

Step 2

Apply CSS to the clock and the hand image so the hand is centered in the clock. It’s a bonus if you make the hand spin with the animate property.

Step 3

Apply CSS to the table so the odd and even rows look different, pretty and distinguishable.

Module 6: Dynamic elements using JavaScript (DOM)

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Design and build a calculator

Step 1

Using HTML elements build a calculator. You will need at least one input field and buttons.

Step 2

Apply CSS styles to make the calculator seem like a calculator. It's up to you how the calculator is going to look and feel.

Task 3 Create the logic of the calculator

Step 1

Create an external script file or open a script tag in the html document.

Step 2

Use the document.getElementById() function to select the nodes of the DOM that corresponds to the screen of the calculator and the buttons and store them in a variable.

Step 3

Create a listener that adds a number to the screen each time you click on a number button. Do the same for all numbers.

Step 4

Create four functions, one to add, one to substract, one to multiply and another to divide. All functions will receive 2 parameters and return one value.

Step 5

Choose the best approach to call your functions from the calculator. You can check on the solutions folder for one approach.

Module 7: Communicate to a remote server

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Add user interaction elements

Step 1

Create a buttons with the “Get fact” text.

Step 2

Add an id property to the button.

Step 3

Create an external script file or open a script tag in the html document.

Task 3 Add functionality

Step 1

Add listeners to detect interaction with the button and test they’re working with an alert. Proceed to next step if the button is working.

Step 2

Add a function so it can retrieve data from your API.

Step 3

Add the code so that each time you press the button you call the function that retrieves the API.

Step 4

Load the data contained in the response into the element with the api_content id.

Module 8: Login Pages

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Local DB

Step 1

Create an external javascript file to store the credentials of a new user. This is going to simulate a local DataBase.

Task 3 Login Form

Step 1

Create the sign in form with the needed fields according to the data on the database. For example, if you stored a username, an email and a password you could use either the username or the email and the password in your login in form.

Step 2

Make sure to use the appropriate input types.

Step 3

In an internal tag or an external script, create javascript variables to access the input's values and create a listener for the submit button.

Step 4

Using the input's values compare the data captured with the data in the local DB, if the strings match then send a message of login successful, otherwise send the message "Wrong credentials or user not found.".

{ } CHALLENGE

Lab: Basic HTML Elements

Task 1: Build a basic HTML structure

Step 1

Insert the !doctype, html, head and body tags into a new html5 document.

Task 2 Choose a theme

Step 1

Open the “Design requirements” file and review the theme options available and choose one.

Step 2

Continue reading the design and functionality.

Step 3

Mockup the main page using HTML5 elements.

Step 4

Apply CSS styles to the elements so they look and behave as stated in the design requirements file.

Task 3 Add functionality

Step 1

Create a script file and code the signup and login functionality by storing new users in an array of users.

Step 2

When you sign in a user, make it so that their credentials are stored using the javascript’s sessionStorage local storage option.

Step 3

If the user is signed in modify the header so it displays a default profile picture.

Step 4

Add the catalog products to the page making use of the appropriate tags.

Step 5

Make it so only logged in users can see the available catalog, and other visitors can only see a big poster and some text explaining the contents of the website if they sign up.

Step 6

Make sure to create all the content specified in the Design requirements document.