**Introduction to HTML and CSS - Exercises**

Problems for exercises and homework for the ["HTML and CSS" course @ SoftUni](https://softuni.bg/courses/html-and-css).

# Topics Covered

* **What is HTML?**
* **HTML Syntax**
* **Tags:**
  + **html, head, title, meta, body,**
  + **h1, h2, h3, h4, h5, h6, p, a, strong, em, ul, ol, li, dl, dt, dd, img, blockquote**
  + **main, aside, article, section, header, footer**
  + **details, summary, figure, caption**

# Resources

* **HTML Element references on MDN:**
  + <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
* **HTML Basics**
  + <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics>
* **Semantics:**
  + <https://developer.mozilla.org/en-US/docs/Glossary/Semantics>
* **W3 Validator:**
  + <https://validator.w3.org>

## Simple image gallery

Build a simple image gallery using HTML and CSS to display a collection of images.

Картина, която съдържа екранна снимка, изкуство

Описанието е генерирано автоматично

### Requirements

* **HTML Document Setup**:
  + Create an HTML document named **image-gallery.html**.
  + Include a **<div>** container to hold the image gallery.
* **CSS File Setup**:
  + Create a CSS file named **gallery-styles.css**.
  + Link the **gallery-styles.css** file to your **image-gallery.html** document using the **<link>** tag in the **<head>** section.
* **Gallery Structure**:
  + Inside the gallery container, use **<img>** tags to include the 12 provided images.
  + Optionally, wrap each **<img>** tag in an **<a>** tag so that the images can be clicked to view the full size in a new tab.

## ToDo List

Build a simple ToDo List using HTML and CSS.

Картина, която съдържа текст, екранна снимка, бял, дизайн

Описанието е генерирано автоматично

### Requirements

* **HTML Document Setup**:
  + Create an HTML document named **todo-list.html**.
  + Include a basic HTML structure (**<!DOCTYPE html>**, **<html>**, **<head>**, **<body>**).
* **CSS File Setup**:
  + Create a CSS file named **todo-styles.css**.
  + Link the **todo-styles.css** file to your **todo-list.html** document using the **<link>** tag in the **<head>** section.
* **Structure**:
  + The page should have a heading (**<h1>**) that says "My ToDo List".
  + Include an input field for adding new tasks.
  + Add a button next to the input field labeled "Add" which, for the sake of this task, will aesthetically complement the input field.
  + Display tasks in an unordered list (**<ul>**). Each task should be in its own **<li>**.
* **CSS Styling**:
  + Style the heading, input field, and button for a cohesive, attractive appearance.
  + Tasks in the list should be styled to be easily readable. Consider font size, item spacing, and list aesthetics.
  + Provide visual feedback for tasks that are marked as done. This could involve changing the color of the task, striking it through, or changing its opacity.
* **Interactivity (CSS Only)**:
  + Though functionality for adding tasks or marking them as done typically requires JavaScript, for this task, demonstrate a simple CSS approach to simulate marking tasks as done. Use the **:hover** pseudo-class to change the appearance of list items when hovered over, indicating they can be interacted with.

## Navigation Bar

Build a simple navigation bar using HTML and CSS that includes links to different sections of a website.



### Requirements

* **HTML Document Setup**:
  + Create an HTML document named **navigation-bar.html**.
  + Include the basic HTML structure (doctype, html, head, body).
* **CSS File Setup**:
  + Create a CSS file named **navigation-styles.css**.
  + Link the **navigation-styles.css** file to your **navigation-bar.html** document using the **<link>** tag in the **<head>** section.
* **Navigation Bar Structure**:
  + The navigation bar should be contained within a **<nav>** element.
  + Include an unordered list (**<ul>**) within the **<nav>** element. Each item in the list (**<li>**) represents a link (**<a>**) to a section of the website (e.g., Home, About, Services, Contact).

## List of blog posts

Develop a simple webpage that displays a list of blog posts, each with a title, an excerpt, and meta information including the post time and author.

Картина, която съдържа текст, екранна снимка, Шрифт, дизайн

Описанието е генерирано автоматично

### Requirements

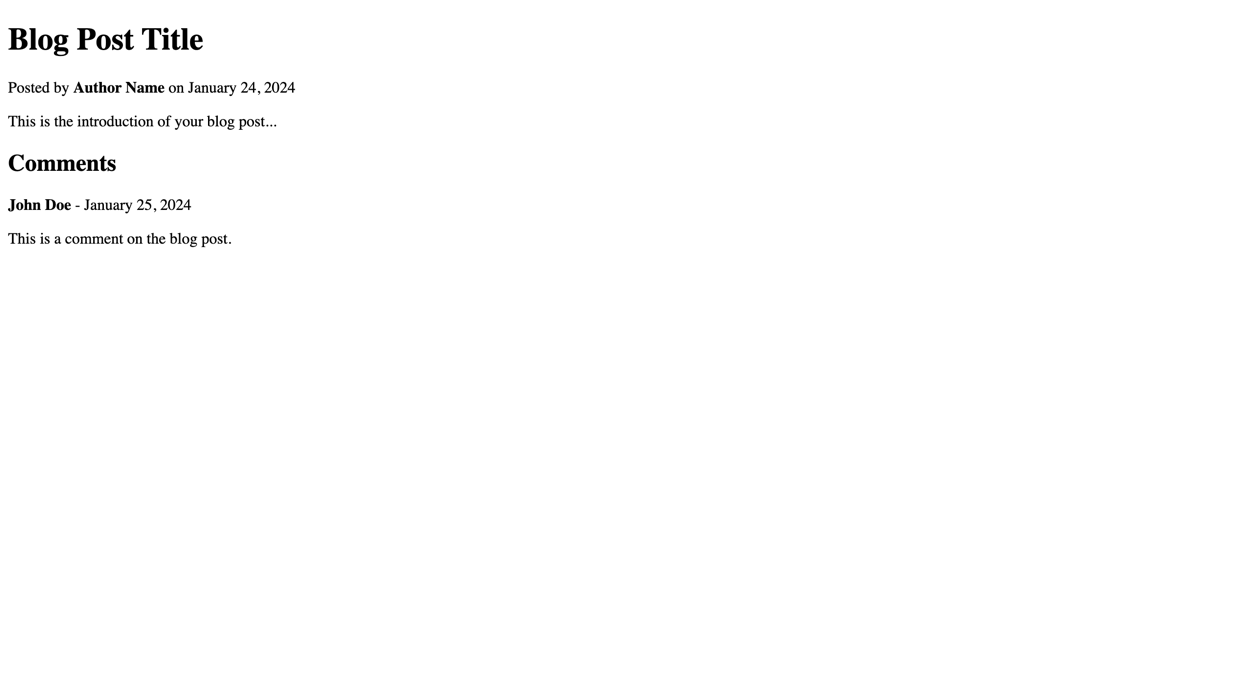
* **HTML Document Setup**:
  + Create an HTML document named **list-of-posts.html**.
  + Ensure the document includes the basic HTML structure (doctype, html, head, body).
* **CSS File Setup**:
  + Create a CSS file named **list-of-posts.css**.
  + Link the **list-of-posts.css** file to your **list-of-posts.html** document using the **<link>** tag in the **<head>** section.
* **Blog Posts Structure**:
  + Use an **<article>** element for each blog post to semantically represent the content.
  + Within each **<article>**, include a **<header>** element that contains the post title (**<h2>**), post time (**<time>**), and author (**<span>** or **<p>**).
  + Follow the header with a paragraph (**<p>**) serving as the post excerpt or summary.

### Resources

* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/blockquote>

## Single blog post page with comments

Create a single blog post page that displays the post content along with a comments section below it.



### Requirements

* **HTML Document Setup**:
  + Create an HTML document named **single-post.html**.
  + Include the standard HTML structure (**<!DOCTYPE html>**, **<html>**, **<head>**, **<body>**).
* **CSS File Setup**:
  + Create a CSS file named **post-styles.css**.
  + Link the **post-styles.css** file to your **single-post.html** document using the **<link>** tag in the **<head>** section.
* **Blog Post Structure**:
  + Use semantic HTML to structure the blog post. This should include a **<header>** with the post title (**<h1>**) and metadata such as the author and publish date (**<p>**).
  + Include a **<main>** section for the blog post content, utilizing **<p>** for paragraphs, **<h2>** for subheadings, and other relevant tags as needed for the content (e.g., **<img>** for images, **<blockquote>** for quotes).

### Resources

* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p>

## Blog layout

Create a responsive blog layout with a main content area for blog posts and a sidebar for supplementary content, using HTML and CSS.

A white background with black dots

Description automatically generated

### Requirements

* **HTML Document Setup**:
  + Create an HTML document named **blog-layout.html**.
  + Include the standard HTML5 structure (doctype, html, head, body).
* **CSS File Setup**:
  + Create a CSS file named **blog-layout-styles.css**.
  + Link the **blog-layout-styles.css** file to your **blog-layout.html** document using the **<link>** tag in the **<head>** section.
* **Blog Layout Structure**:
  + Utilize semantic HTML tags to structure the layout, including **<header>**, **<main>**, **<article>**, **<aside>**, and **<footer>**.
  + The **<header>** should contain the blog’s title or logo and a navigation menu.
  + The **<main>** area should display blog posts, each wrapped in an **<article>** tag, including titles (**<h2>**), publication dates (**<time>**), authors, summaries (**<p>**), and read more links (**<a>**).
  + The **<aside>** should serve as a sidebar, containing widgets or links such as "About Me", categories, recent posts, or social media links.
  + The **<footer>** should contain copyright information and/or links to privacy policy, terms of service, etc.