UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
BACHELOR OF SCIENCE IN COMPUTER SCIENCE
WEB TECHNOLOGIES COURSE – SPRING 2024
TEACHER: Luigi Libero Lucio Starace, Ph.D.

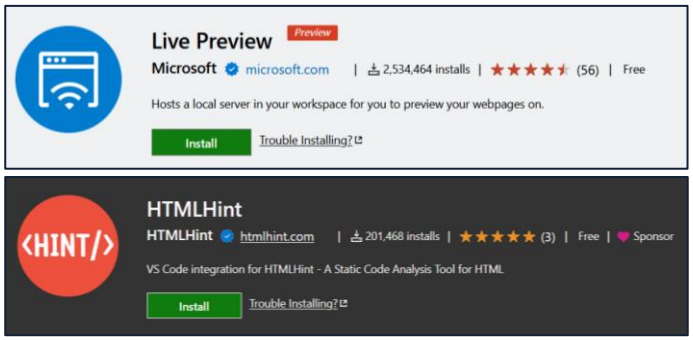
DATE: March 8, 2024
BEST DONE BEFORE: March 12, 2024

ASSIGNMENT: #01 TOPIC: HTML

EXERCISE 1. SETUP YOUR ENVIRONMENT

Download and install the latest version of Visual Studio Code.

Install the Live Preview extension and HTMLHint extension. Extensions can be installed from the official website (click on the links above) or by searching within the Extensions tab in Visual Studio Code (also accessible via CTRL+MAIUSC+X).

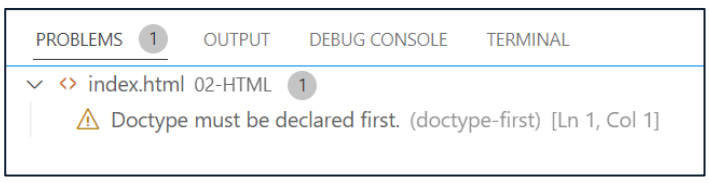


The Live Preview extension hosts a HTTP server within Visual Studio Code, allowing us to preview our web pages. HTMLHint is a Static Analysis tool for HTML. We'll use it to check that our HTML code is up to specification.

Create an empty directory and open it in Visual Studio Code.

Create a new HTML file within the directory. Be sure to include a `<!DOCTYPE html>` declaration, a `<html>`, a `<head>` with a `<title>`, and a `<body>` with at least one `<h1>`. When you're done, use the Live Preview extension to preview the web page. You can do that by right-clicking on the HTML file on the Explorer tab and selecting "show preview". Feel free to change some parts of your HTML file and see how the webpage in the preview web browser is automatically updated.

When you're done, remove the `<!DOCTYPE html>` declaration at the top of your file. Bring up the Problems tab in Visual Studio Code (CTRL+MAIUSC+M or View >> Problems), and you should notice a warning there (*Doctype must be declared first*).

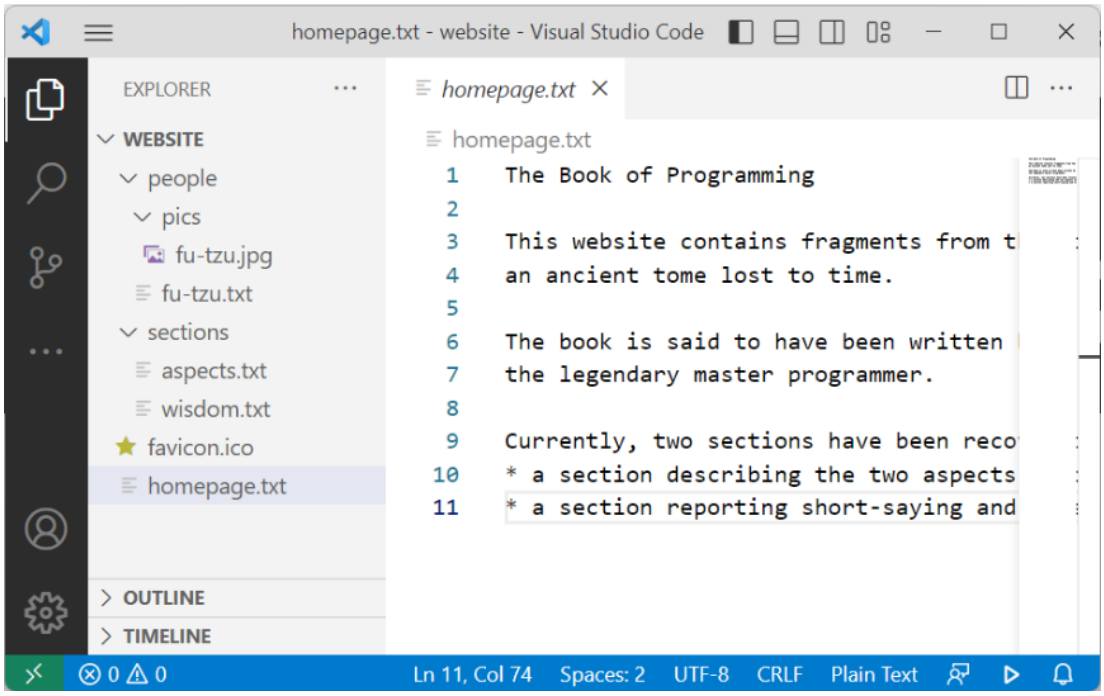


Feel free to introduce other mistakes in the HTML file and check whether the HTMLHint extension is picking them up as issues. In the remainder of this assignment, keep an eye on the problems tab to make sure that you're writing syntactically correct HTML documents.

ASSIGNMENT: #01 TOPIC: HTML

EXERCISE 2. YOUR FIRST WEBSITE: THE BOOK OF PROGRAMMING

Download the support materials provided in `website.zip` and extract the zip archive in a new directory. Open the new directory in Visual Studio Code.



As you can see from the Explorer tab in Visual Studio Code, the base materials consist in four txt files organized in directories. Start by renaming each `txt` file, changing its extension to `.html`. Subsequently, edit the files and include markup to make them proper HTML files. Feel free to use all the tags you believe are appropriate, and to add content, if you want to. Try to group elements using semantic tags.

The `homepage.html` document should include links to all other HTML documents on the website. All other documents should include at least a link going back to the homepage. You should use relative URLs for all links.

Start the Live Preview web server and navigate your website. Make sure that all links are working correctly and you get no 404s! And remember to keep an eye out for problems in the HTML syntax.

Take good care of this very first website you developed during the Web Technologies course, we might get back to it in future assignments!

ASSIGNMENT: #01 TOPIC: HTML

EXERCISE 3. FORMS

Write a new HTML document to allow users to sign up for a certain service. The HTML document will contain a form with at least the following inputs:

- First name and last name
- Birth date
- Radio buttons to select a gender (male, female, other, do not declare)
- Desired password
- A short biography (use a `<textarea>` input)
- The favourite color (use `<input type="color">`)

Use `labels`, `fieldset` and `legend`. Use `method="GET"` on the form.

When you are done with the HTML document, answer the following:

- What happens when you submit the form? What is the query string?
- If you change the form method to `"POST"`, what happens when you submit the form? Is there a query string?
- Use the browser's web tools to inspect the POST request that is sent upon submission of the form. What is the body of the POST request?
- What happens, upon submission, if one of the inputs does not have a name attribute?

ASSIGNMENT: #01 TOPIC: HTML

EXERCISE 4. USING A PRODUCTION-GRADE HTTP SERVER (★)

Install a production-grade HTTP server (e.g.: Apache HTTPd or NGINX) on your machine and use it to serve the website you developed in Exercise 2.

We'll do that using NGINX and Docker. We start by looking at the official NGINX image. The docs in this page include many nice examples of how to get NGINX up and running in a Docker container, and on how to provide it with our website to serve.

We will create a custom NGINX image, in which we copy the custom website we developed in the default NGINX document root (which is the directory `/usr/share/nginx/html`). Assuming that the website we developed is in a local `./website` directory, we create a `Dockerfile` in the same directory containing `./website`, with the following content:

```
# Start from a base nginx image
FROM nginx:1.25.4

# Copy the book of programming website directory into /usr/share/nginx/html
COPY ./website /usr/share/nginx/html
```

We can now build a new Docker image from the above Dockerfile, and then run the new Docker image as shown below.

```
@luigi ➜ solution $ docker build -t book-of-programming-nginx-image .
@luigi ➜ solution $ docker run --name book-of-programming-nginx-container
-p 80:80 -d book-of-programming-nginx-image
```

The website should be up and running at http://localhost/homepage.html. Once you are done inspecting the website, stop the running container and (optionally) delete the image you build using the following commands:

```
@luigi ➜ solution $ docker container stop book-of-programming-nginx-container
@luigi ➜ solution $ docker container rm book-of-programming-nginx-container
@luigi ➜ solution $ docker image rm book-of-programming-nginx-image
```

Feel free to replicate the above steps using a different HTTP server, such as – for example – Apache HTTPd. The steps will be roughly the same.

If you are interested in learning more about HTTP servers (not required for the Web Technologies course!), you can start by looking at the docs (e.g.: https://nginx.org/en/docs/).