

Web Development and Databases

Practical 2 – HTML

Practical Aim

- To learn and create simple web pages using HTML.
- To learn and use different HTML tags.
- To create well-formed HTML web pages and validate it.

Pre-requisite:

- You have a web browser to load/render your web pages.
- You have VS Code or any other suitable editor to write HTML script.
- You have already created a workspace directory on your computer where you can save all your scripts. Please note: H:/ is shared drive and you should be able to access it from different lab computers.
- You have gone through lecture slides / recording.

Post-requisite:

- Make sure you take backup of your work.

A quick guide:

A good guide to follow is accessible from http://www.w3schools.com/html/html_intro.asp. Most important HTML element you must practice are:

- <Doctype>, <html>, <head>, <title>, <body>, <a>, <table>, , <p>, <h1>, <iframe>, <h2>, ,
, , , HTML entities, HTML colours, <header>, <nav>, <section>, <article>, <footer>, <aside>
- HTML forms - http://www.w3schools.com/html/html_form_input_types.asp with most input types including 'text', 'password', 'submit', 'radio', 'checkbox', 'button', 'color', 'date', 'email', 'number', 'range', 'url', 'time', etc
- Also work on input restrictions e.g. disabled, max, maxlength, min, readonly, required, size, step, value etc.

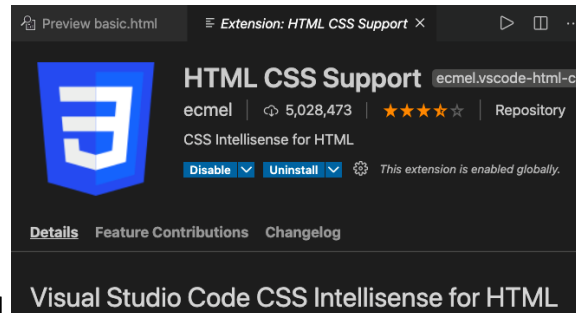
Note: If you need help - **ask**. If you are curious - **ask**. If you want to know - **ask**. You may work in groups but this is not compulsory.

Exercise 1: Making VS Code ready for writing HTML code.

Task 1: You can run your HTML files in a web browser to see the output or you may install the

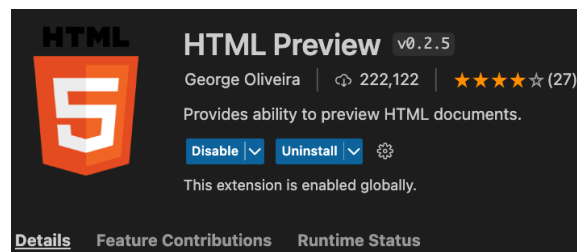


following extensions through extension explorer in VS Code for HTML/CSS and previewing output. Your tutor will demonstrate by installing and using one of the following extensions and then try the rest by yourself.



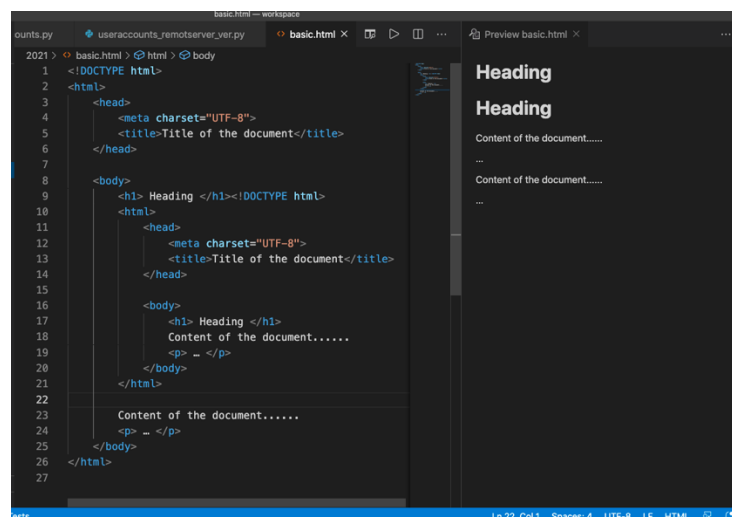
1. HTML CSS Support by ecme1

This will present you with different options when you'll start typing a <tag> in VS Code.

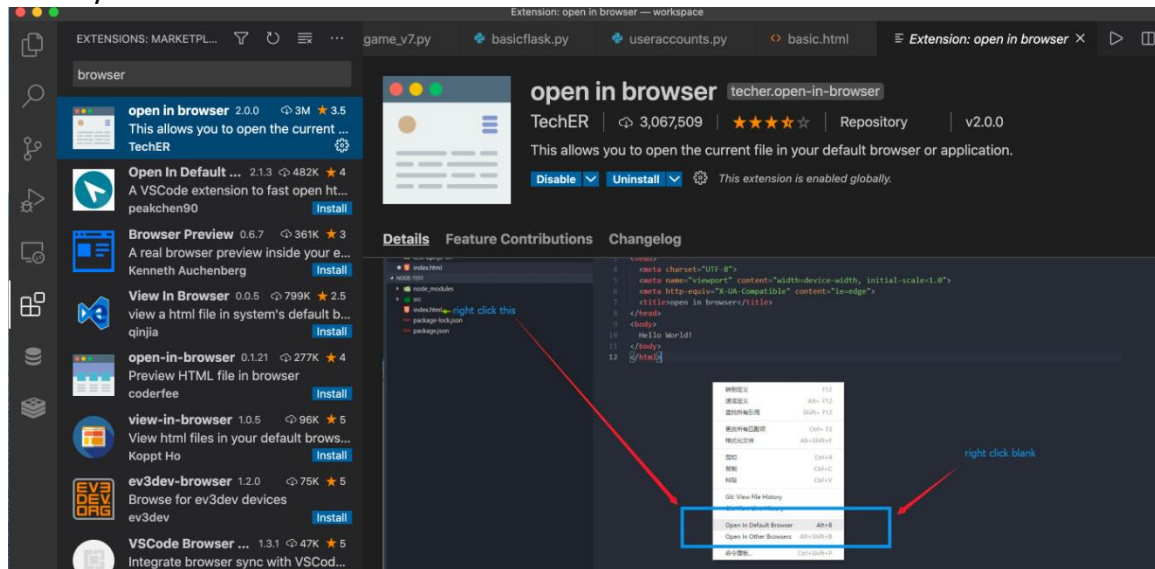


2. HTML Preview by George Oliveira

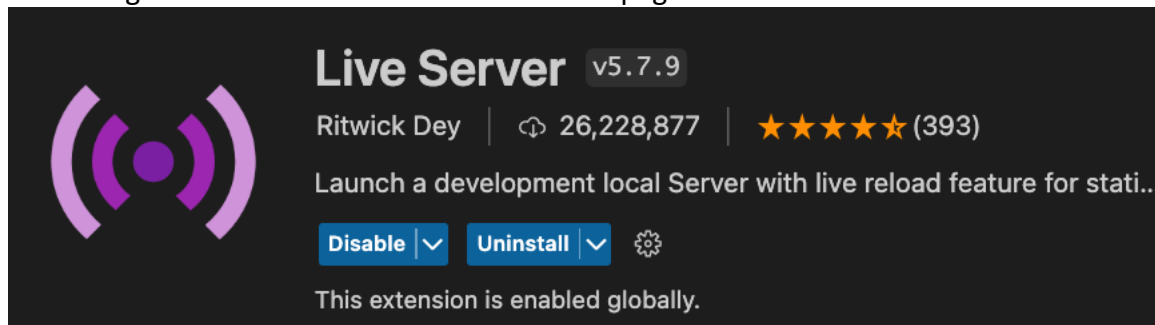
With HTML Preview you can preview the output of your HTML tags. It will look something like this in your VSCode editor. It only shows the layout of your page.



- You may also use extension 'open in browser' to load your webpage in a Web browser directly from VSCode.



- You may also try Live Server by Ritwick Dey. This extension helps to launch a development local server with live reload feature for static websites/web pages. This means when you click on Go Live in status bar (bottom of the VS Code), it opens the .html file in default web browser and if you make any change in the html and save it, this change will be reflected in the loaded web page in the web browser.



So point here is that it's easy to find and install extensions in VS Code and there are hundreds of different extensions you can try. In later practical sessions we'll use more VS Code extensions.

Task 2: From File menu, set up workspace in VSCode. Workspace is a folder on your computer which will be considered as default folder to create and save your program files. You may also open a folder from file menu to make it home directory to save all code files.

On lab computers, can you test whether or not creating a workspace or folder on H: useful? Discuss with your tutor why should you save your files on H:

Exercise 2: Your tutor will demonstrate myFirstHTML.html, first.html and example/second.html

Task 1: Review code of myFirstHTML.html, first.html and example/second.html and try to understand semantics and purpose of all tags used in these files.

Task 2: Then, you need to make a copy of myFirstHTML.html and save it as mySecondHTML.html. Make the following changes in mySecondHTML.html.

1. In HTML table – Kevin Pietersen’s photo is not accessible. Try to find an alternative image from the internet, save it on your disk (in the same folder) and modify the HTML code to show it in the table. **Hint:** look `` tag.
2. If you click on [Local webpage in one of local folder on my computer](#) on the webpage, it will not work!
Find out why is it not working?
Hint: look `<a>` tag.
3. Replace `<a>` tag href attribute value "c:\myweb\web.html" with a working html page from your computer.
Hint: look `<a>` tag and to make it work, create another webpage if there is no other webpage on your computer.
4. For form elements: try different restrictions, e.g. required, size, maxlength, readonly etc. see list here: https://www.w3schools.com/html/html_form_attributes.asp
Hint: check data input validation e.g., date range, number range, min, max, etc.
5. Convert unordered list `` with an ordered list ``
Hint: check HTML ol element
6. Make the HTML table line invisible. **Hint:** check border attribute
7. Create a folder temp and create third.html inside the temp folder. Now update third.html and create bi-directional hyperlinks with first.html. For example, you should be able to go to third.html from first.html and come back to first.html from third.html. You should create similar bi-directional links between second.html and third.html. **Hint:** check how first.html and second.html are linked. Also, check Exercise 3.

Exercise 3:

HTML 5 has introduced new elements (or tags) with better semantics so that you can structure your webpage contents properly. For example, HTML 5 elements

http://www.w3schools.com/html/html5_semantic_elements.asp
like `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>` etc.

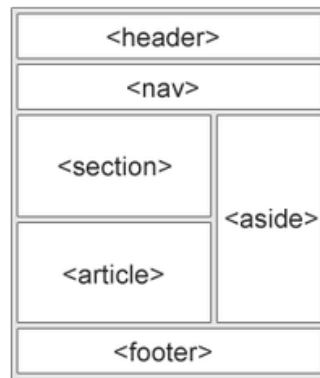
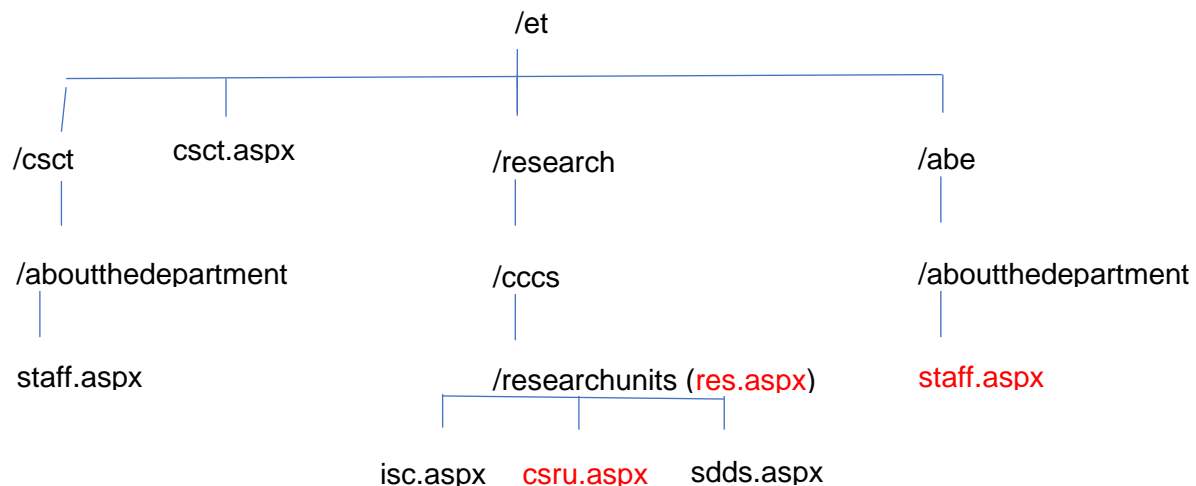


Figure 1 : HTML5 tags

Use the **UWEhtml4.html** and apply above HTML5 elements as appropriate.

Exercise 4: Let's assume we have the following site map:



Indicative examples of absolute URL addresses from the above sitemap can be something like:

<http://www1.uwe.ac.uk/et/csct/aboutthedeartment/staff.aspx>
<http://www1.uwe.ac.uk/et/research/cccs/researchunits/isc.aspx>
<http://www1.uwe.ac.uk/et/abe/aboutthedeartment/staff.aspx>

Now assume that we are on the **csru.aspx** webpage.

Task 1: Write the <a> tag with relative address to go one level up to hyperlink **res.aspx**

Task 2: Write the <a> tag with relative address to go four level up to access **staff.aspx** of the /abe department.

Exercise 5: Table_example.html shows how colspan can be used to merge cells. Perform the following tasks:

Task 1: Add one more column.

Task 2: Can you merge cells in two adjacent rows? If yes, they try to merge row 3.col 4 and row 4. col.4. **Hint:** try rowspan attribute.

Exercise 6: Contents and sitemap for your Website project

Task 1: In this exercise spend some time to think about different web pages you would like to have in your website project and how will these pages be linked with each other. Consider drawing a sitemap. This is your opportunity to start working on your Website project. Don't worry if you later change your mind and come up with a completely different sitemap. You may also start thinking about structuring your contents on each web page (e.g., wireframes). You can start adding static contents for your pages e.g., privacy statement, hotel intro, about us etc.

Task 2: Identify places or contents in your web pages which you would like to replace with dynamic contents. This is like creating a template and putting placeholders for dynamic contents such as user profile, booking information etc.

Exercise 7: For more practice you can create a static website. Discuss with your tutor what is a static website.

To challenge yourself you can work on a small problem specification. In this exercise, you are expected to create a student enrolment website. This means you should create three or more linked web pages and each web page has some useful information about modules you're studying. You need to be creative for this exercise. For example, you may choose Figure 2 design as your front end design. You need to demonstrate use of different HTML elements and attributes.

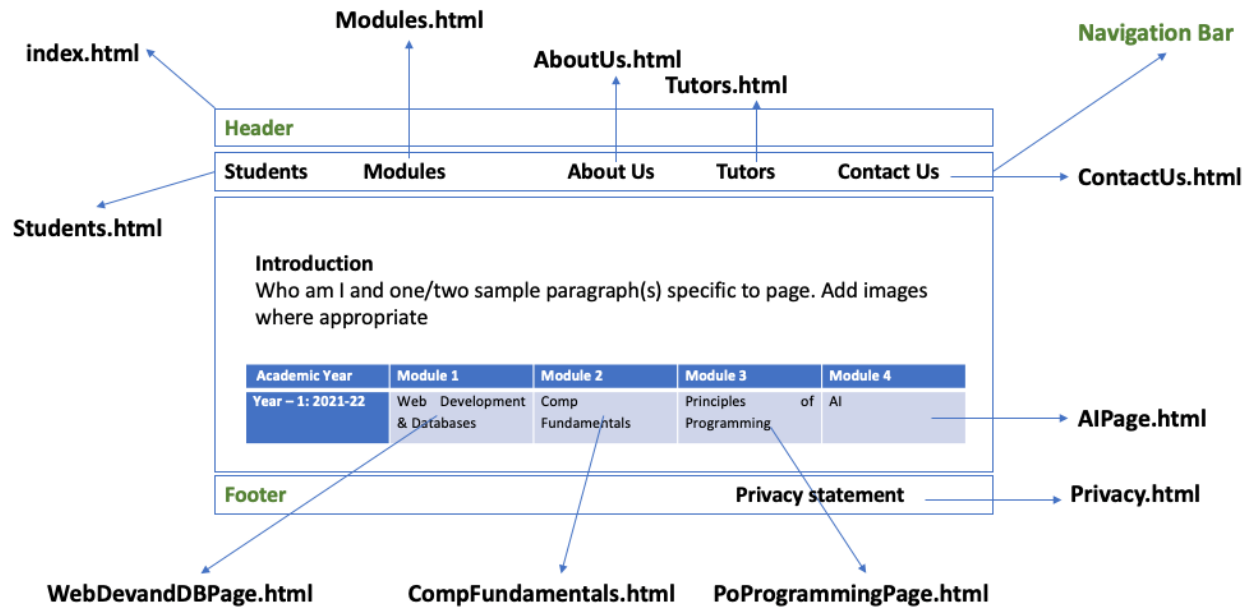


Figure 2. Student enrolment website

Be creative and you may use the following **hints** to complete your website pages.

Anchors <a>

- Use <a> tag to create navigation menu that should allow you to navigate through different web pages on your website.
- You may like to use <a> tag to link your web pages to course website (from UWE website).
- You may like to use <a> tag to link your course to external resources e.g. tutorials, ebooks, articles, etc.

Tables <table>

You can create main or index.html page and use a table to show all of your modules, the module name and any other information (what you think of the module or related books or online resource, for example.)

- Experiment with the <table> <td><tr> tags and their attributes.
- You will need to get proficient at HTML tables

Images

Use images as appropriate e.g., UWE image, logo etc.

You can get images from any website which has them by using the Right Mouse Button

and **Save Image As ...** save the image into a sub-directory **images** (you'll need to create **images** directory). Also do **check legal and ethical issues e.g. copyrights for downloading and using images on your website!**

You should make all of the images appear to be the same size using the **height** and **width** attributes of **** tag. You should use % values for height and width for responsiveness.

Lists, Headings, Line breaks, colours and Paragraphs

Use lists (ordered `` or unordered ``) to show what topics you have covered for each module

Use appropriate heading `<h1>`, `<h2>` etc where needed

Use paragraphs `<p>` where needed

Use line breaks using `
`

Try to have different background colour for each page.

Forms

Create a contact us page and try to use different form elements with different properties and restrictions. You should at least have following items on the contact us form page:

- Name with a text field
- Email with a text field
- Comments with a text area
- Use a list for selection of predefined items
- Use checkboxes for selection of predefined items
- Use radio buttons for selection of predefined items
- Submit button

Note: Forms are very important to send client data to server script. We'll use them with Python Flask. So you should spend good time to learn different form elements and constraints.