

Lecture/Tutorial/Laboratory 04 (SDL)

Web Validation and Accessibility Testing

1. LEARNING OUTCOMES

Upon completion of this lesson, you should be able to:

- Validate HTML content using the W3C Markup Validation Service.
- Evaluate the accessibility of a website using the axe accessibility testing toolkit.

2. REQUIRED SOFTWARE

- Apache NetBeans 11.2 (or later):
<https://netbeans.apache.org/download/index.html>
- FireFox (<https://www.mozilla.org/en-US/firefox/new/>) or Chrome (<https://www.google.com/chrome/>) web browser.
- W3C Markup Validation Service: <https://validator.w3.org/>
- Deque's [axe](#) accessibility testing plugin
 - Firefox: <https://addons.mozilla.org/en-US/firefox/addon/axe-devtools/>
 - Chrome: <https://chrome.google.com/webstore/detail/axe-web-accessibility-tes/lhdoppoipmngadmndnejejpokejbdd>

3. INSTRUCTIONS

- 3.1 This document combines the Lecture, Tutorial and Laboratory for Week 4, and is to be completed on your own via self-directed learning (SDL).
- 3.2 Study the reference materials and online resources, then complete the exercises as instructed. Take note that you will be responsible for knowing this material for the Quiz/Exam. Additionally, you must submit your work to xSiTe in order to receive credit for the assignment.

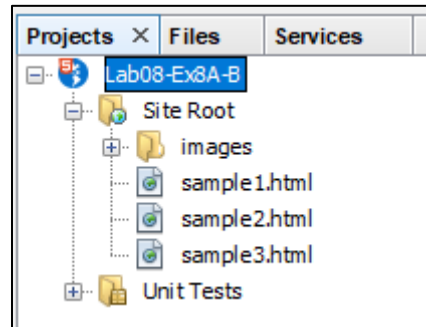
4. STUDY REFERENCES

- 4.1 Study the topics on W3C under **Introduction to Standards-Based Design**, especially the use of W3C's HTML validator:
<https://www.w3.org/standards/about.html#started>.
- 4.2 Review the **Standards FAQ** at: <https://www.w3.org/standards/faq.html>.
- 4.3 Study **Introduction to Web Accessibility**:
<https://www.w3.org/WAI/fundamentals/accessibility-intro/>.
- 4.4 Study the Web Content Accessibility Guidelines (WCAG):
<https://www.w3.org/WAI/standards-guidelines/wcag/>. In particular, **be sure** you know the four principles of WCAG, represented by the acronym [POUR](#).

5. LAB EXERCISES - INITIAL SETUP

5.1 Perform the following steps first before completing the exercises:

- Create a new HTML5 project in NetBeans. You can delete the default 'index.html' that's auto-generated, you will not need this file.
- Download the file "Lab04-Sample-Files.zip" from xSiTe and unzip the contents into the 'public_html' folder (Site Root) of your newly created NetBeans project. Your project contents should look like this:



- Install the **axe** plugin using one of the links provided in section 2. The examples are shown in Firefox, but you are welcome to use Chrome if you prefer.

6. EXERCISE 1: WEB VALIDATION

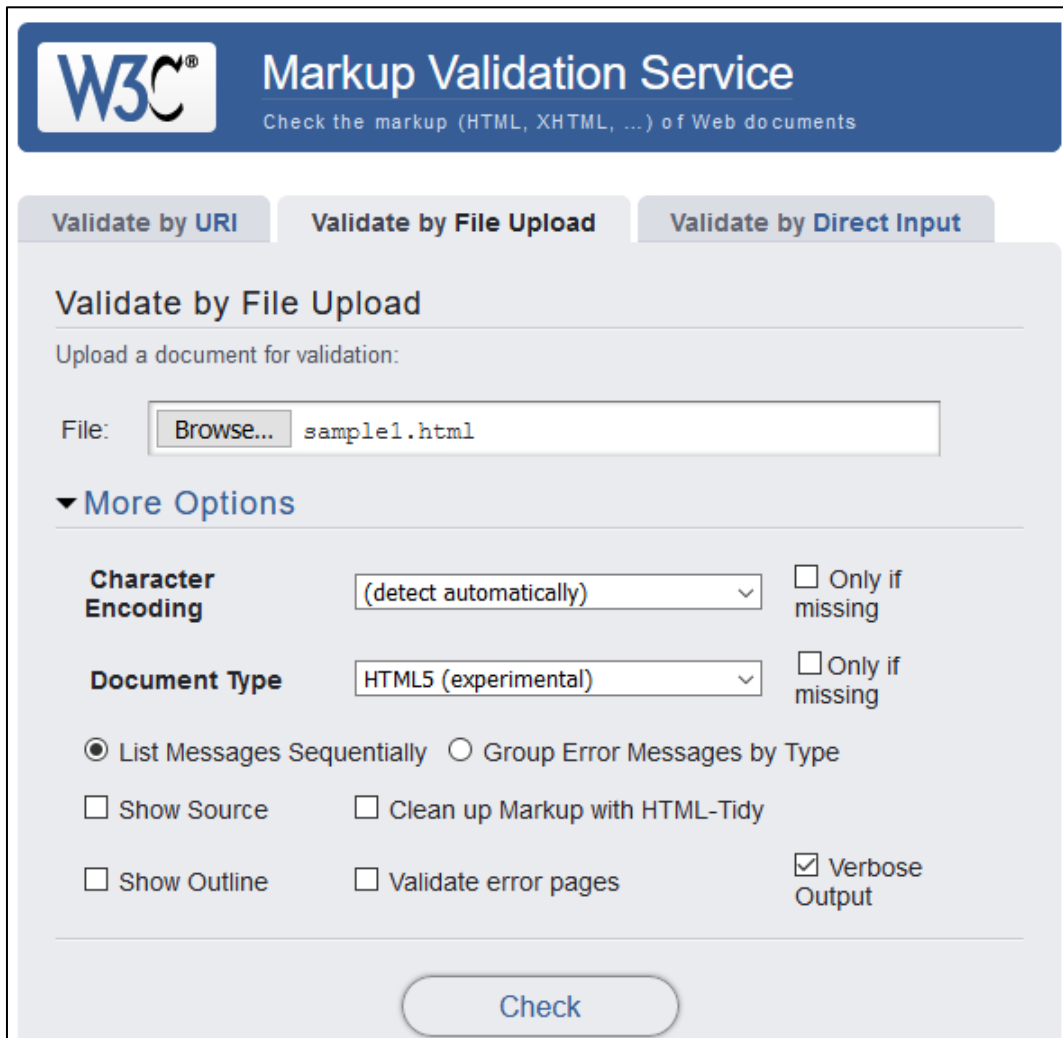
6.1 Let's get some hands-on experience using the W3C Markup Validation Service.

Navigate to <https://validator.w3.org/> and observe the three tabs:

- Validate by URI** - allows you to validate any publicly available web page online. For example, try typing www.singaporetech.edu.sg in the 'Address' box and hit the 'Check' button. You may see some interesting results!
- Validate by File Upload** - allows you to upload a HTML document from your local file system and perform validation. This is the method we'll use in the exercise.
- Validate by Direct Input** - allows you to type or paste HTML content directly in the editor and perform validation.

6.2 Go to the 'Validate by File Upload' tab. Click the 'Browse...' button, navigate to the public_html folder in the NetBeans project you created in 3.1, and select 'sample1.html' as the input file.

Expand the 'More Options' panel and select "HTML5 (experimental)" as the document type. Lastly, check the 'Verbose Output' checkbox to get more explanatory output:



The image shows the W3C Markup Validation Service interface. At the top, there is a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by File Upload" tab is selected. Under this tab, there is a section titled "Validate by File Upload" with the instruction "Upload a document for validation:". Below this, there is a "File:" label and a text input field containing "sample1.html". To the left of the input field is a "Browse..." button. Below the input field, there is a section titled "More Options" with a downward arrow. This section contains several options: "Character Encoding" set to "(detect automatically)" with a dropdown arrow and a checkbox "Only if missing"; "Document Type" set to "HTML5 (experimental)" with a dropdown arrow and a checkbox "Only if missing"; radio buttons for "List Messages Sequentially" (selected) and "Group Error Messages by Type"; checkboxes for "Show Source", "Clean up Markup with HTML-Tidy", "Show Outline", and "Validate error pages"; and a checked checkbox for "Verbose Output". At the bottom of the form is a large "Check" button.

- 6.3 Click the 'Check' button and observe the results. The validator should indicate 1 warning and 2 errors:

1. **Warning** Consider adding a `lang` attribute to the `html` start tag to declare the language of this document.
 From line 1, column 16; to line 2, column 6

```
TYPE html><html><
```


 For further guidance, consult [Declaring the overall language of a page](#) and [Choosing language tags](#).
 If the HTML checker has misidentified the language of this document, please [file an issue report](#) or [send e-mail to report the problem](#).
2. **Error** End tag `header` seen, but there were open elements.
 From line 12, column 13; to line 12, column 21

```
</header><
```
3. **Error** Unclosed element `h1`.
 From line 11, column 17; to line 11, column 20

```
<h1>Explor
```

- 6.4 Edit 'sample1.html' in NetBeans and fix all of the problems. Run the file through the validator again to check that there are no errors:

Document checking completed. No errors or warnings to show.

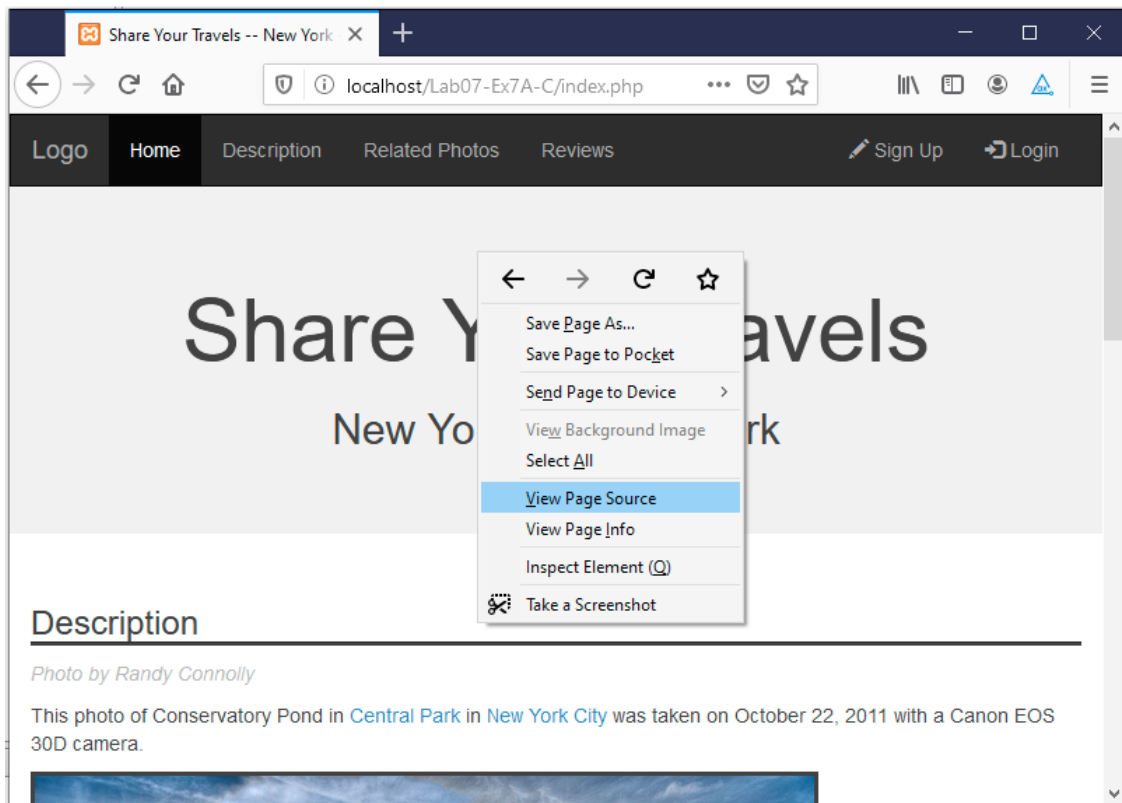
Used the HTML parser.

Total execution time 2 milliseconds.

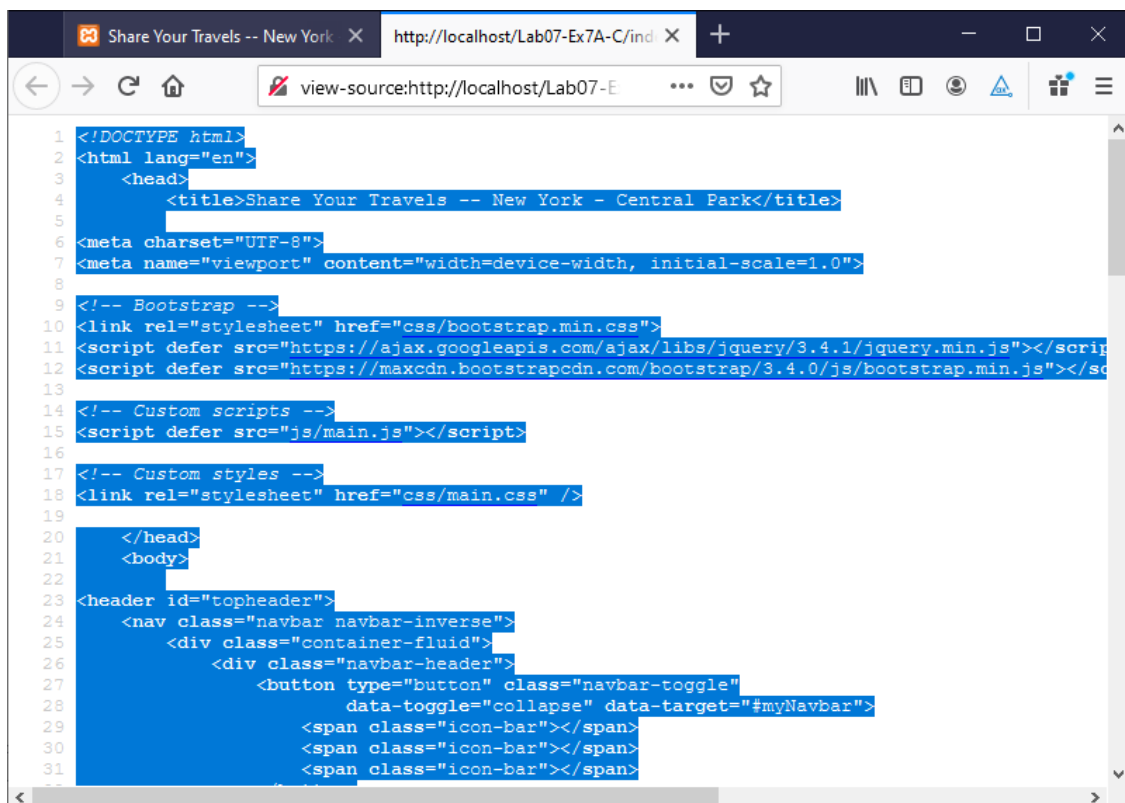
- 6.5 Now repeat the above steps with 'sample2.html' and then 'sample3.html' as the input files. Fix all errors in both files and verify that the updated files pass validation.

7. HOW TO VALIDATE PHP FILES

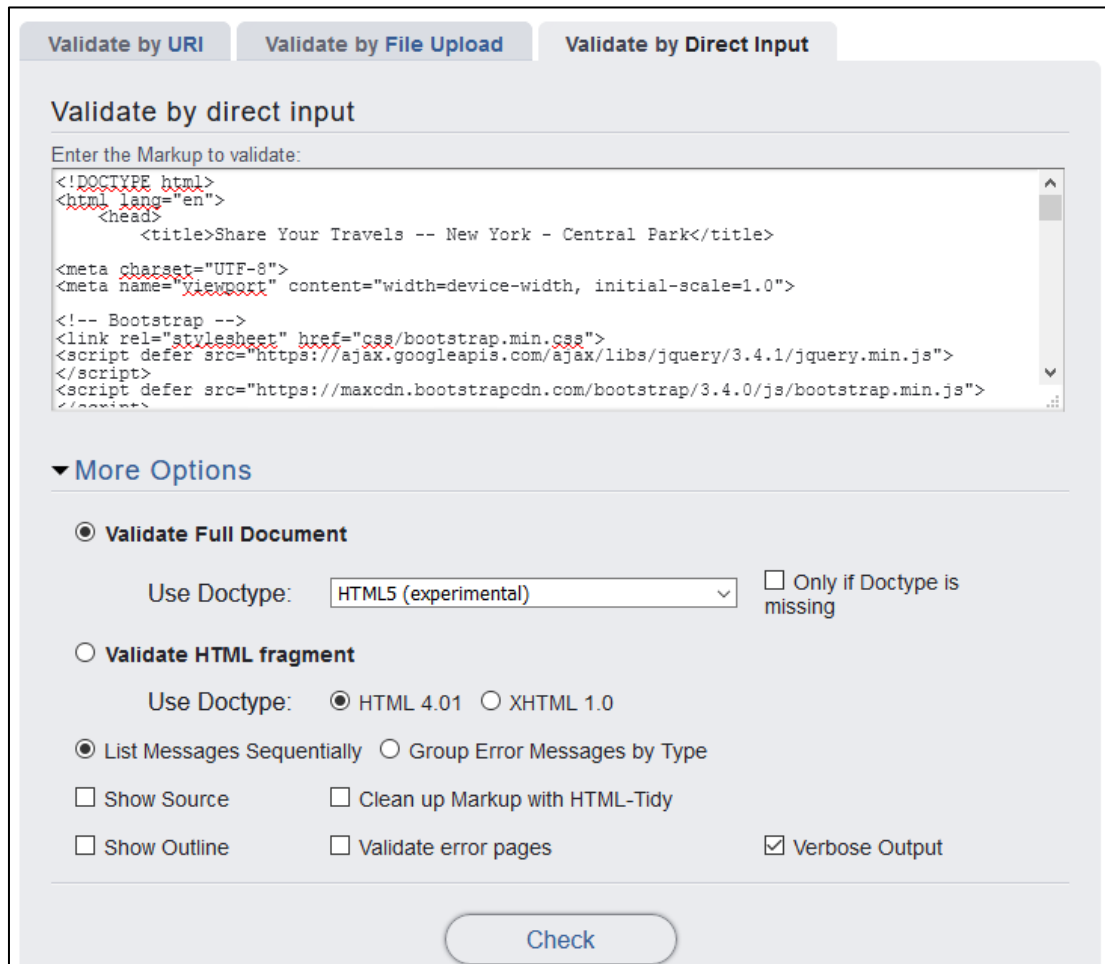
- 7.1 Note that PHP source files **cannot** be uploaded to the W3C Markup Validation Service directly, as we did with the HTML files above. Instead you must use either the 'Validate by URI' method, if your website is publicly accessible, or the 'Validate by Direct Input' method, if your website is running on localhost. To use the latter method, you would:
- a. Run the PHP page on localhost to generate a pure HTML page in the browser.
 - b. Right-click on the page and select 'View Page Source' from the popup menu:



- c. Select all of the HTML code in the source view (CTRL-A) and copy to the clipboard (CTRL-C):



- d. Go to the 'Validate by Direct Input' tab in the W3C Markup Validation Service and paste (CTRL-V) the content into the editor:



Validate by URI Validate by File Upload **Validate by Direct Input**

Validate by direct input

Enter the Markup to validate:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Share Your Travels -- New York - Central Park</title>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Bootstrap -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script defer src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
    </script>
    <script defer src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js">
    </script>
```

▼ More Options

☒ **Validate Full Document**

Use Doctype: HTML5 (experimental) ▼ ☐ Only if Doctype is missing

☐ **Validate HTML fragment**

Use Doctype: ☒ HTML 4.01 ☐ XHTML 1.0

☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☒ Verbose Output

Check

- e. Click 'Check' to perform the validation and view the results:

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report Options...

Check by text input ☐ CSS

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Share Your Travels -- New York - Central Park</title>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Bootstrap -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script defer src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script defer src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

    <!-- Custom scripts -->
    <script defer src="js/main.js"></script>
          
```

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

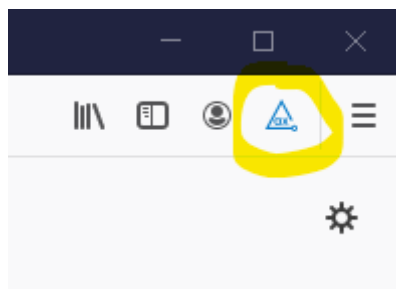
Document checking completed. No errors or warnings to show.

8. EXERCISE 2: ACCESSIBILITY TESTING

- 8.1 The validation tests performed in exercise 1 will help you to correct HTML5 content problems, however they do not help very much with accessibility issues. For this, we need to conduct additional testing using a dedicated accessibility evaluation tool.

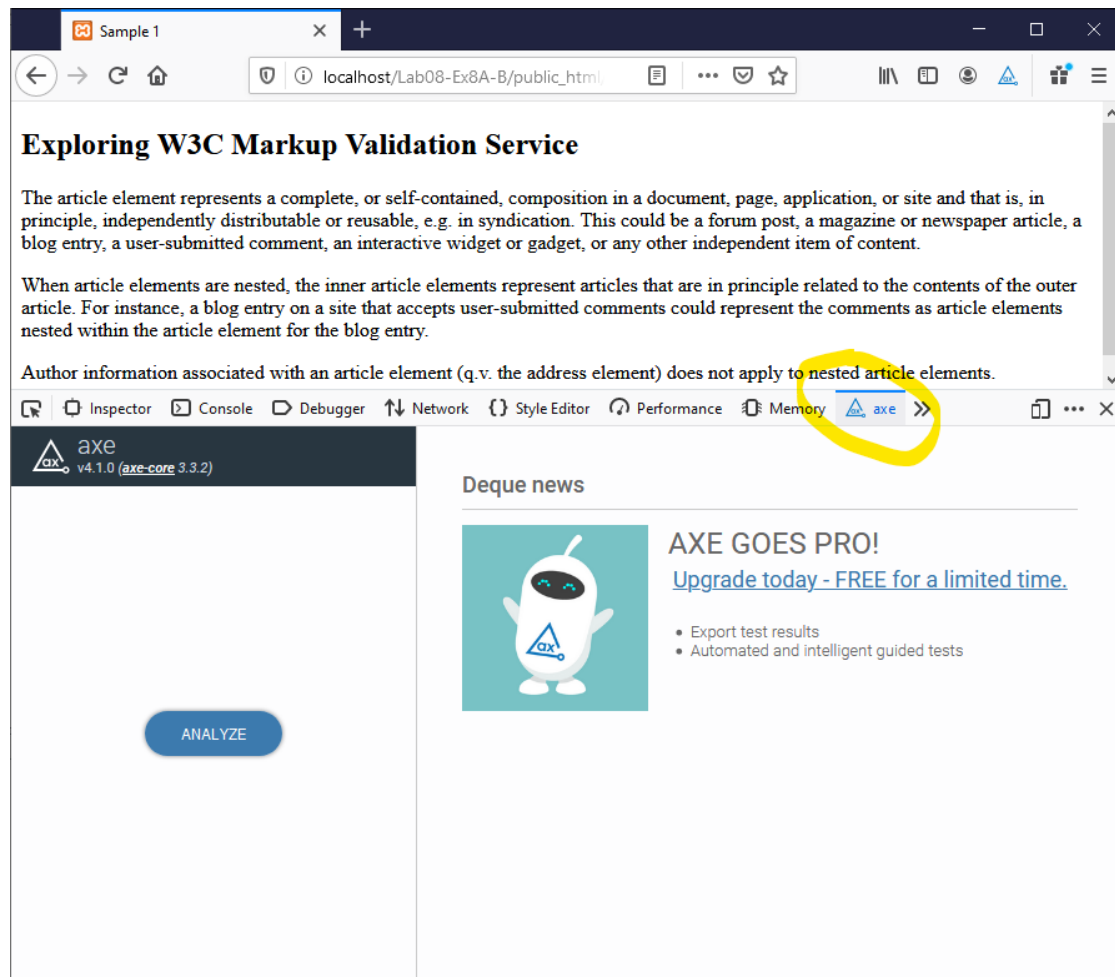
There are many such tools available (refer to <https://www.w3.org/WAI/ER/tools/>) and you are encouraged to experiment, however for this exercise we are going to use **axe**, one of the most popular accessibility testing tools available.

If you successfully installed either the Firefox or Chrome extension for axe as directed in section 3, you should see the axe icon in the upper right corner of your browser:

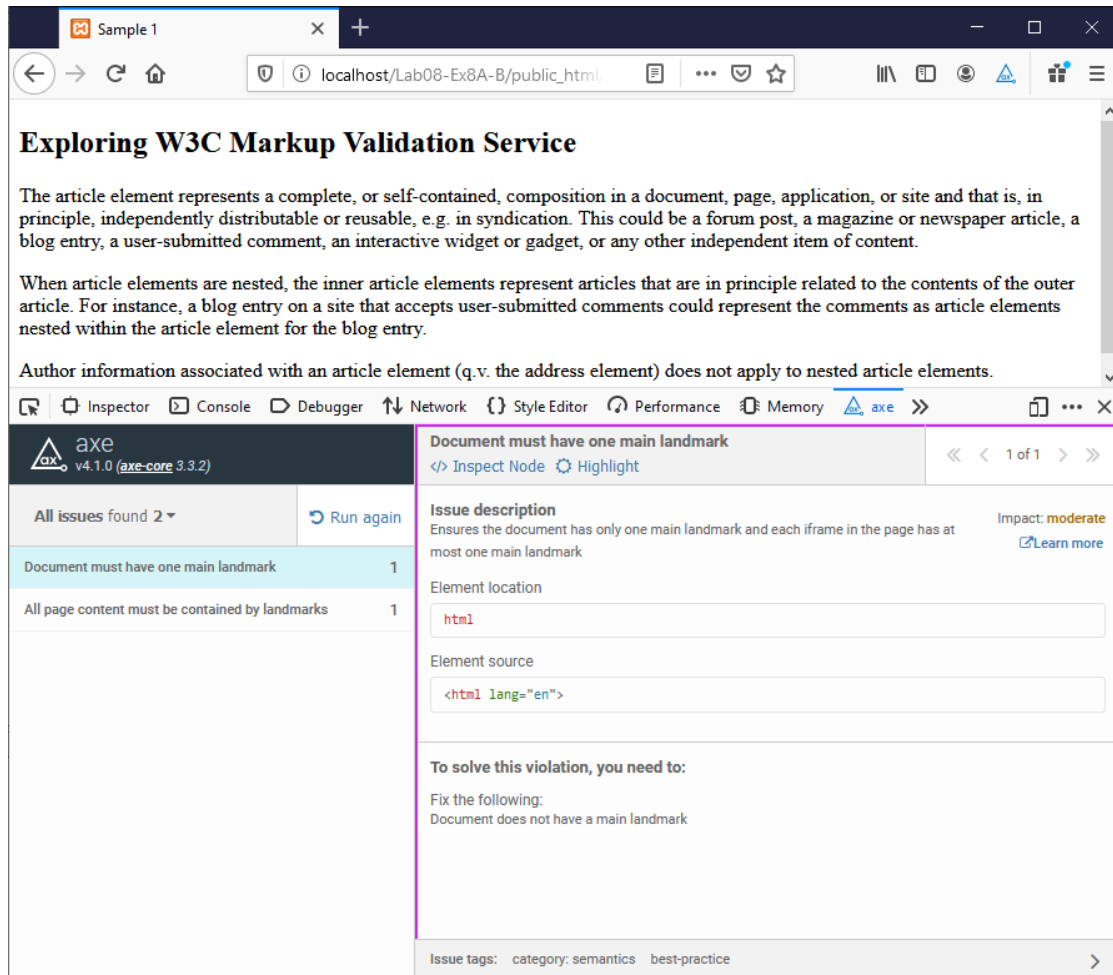


- 8.2 Now we'll perform accessibility testing on each of the sample files that we fixed in exercise 1. Follow these steps, starting with 'sample1.html':

- f. View 'sample1.html' in the browser, then activate the Developer Tools (F12). Look for the axe tab and click it:



- g. Click the 'ANALYZE' button to start the accessibility testing. Two issues are found, both are because we're missing the <main> element in our document:



The screenshot shows a web browser window with the address bar displaying `localhost/Lab08-Ex8A-B/public_html`. The page content includes a heading "Exploring W3C Markup Validation Service" and several paragraphs of text. Below the page content, the browser's developer tools are open, showing the "axe" extension interface. The "axe" extension is version 4.1.0 (axe-core 3.3.2). It shows two issues found:

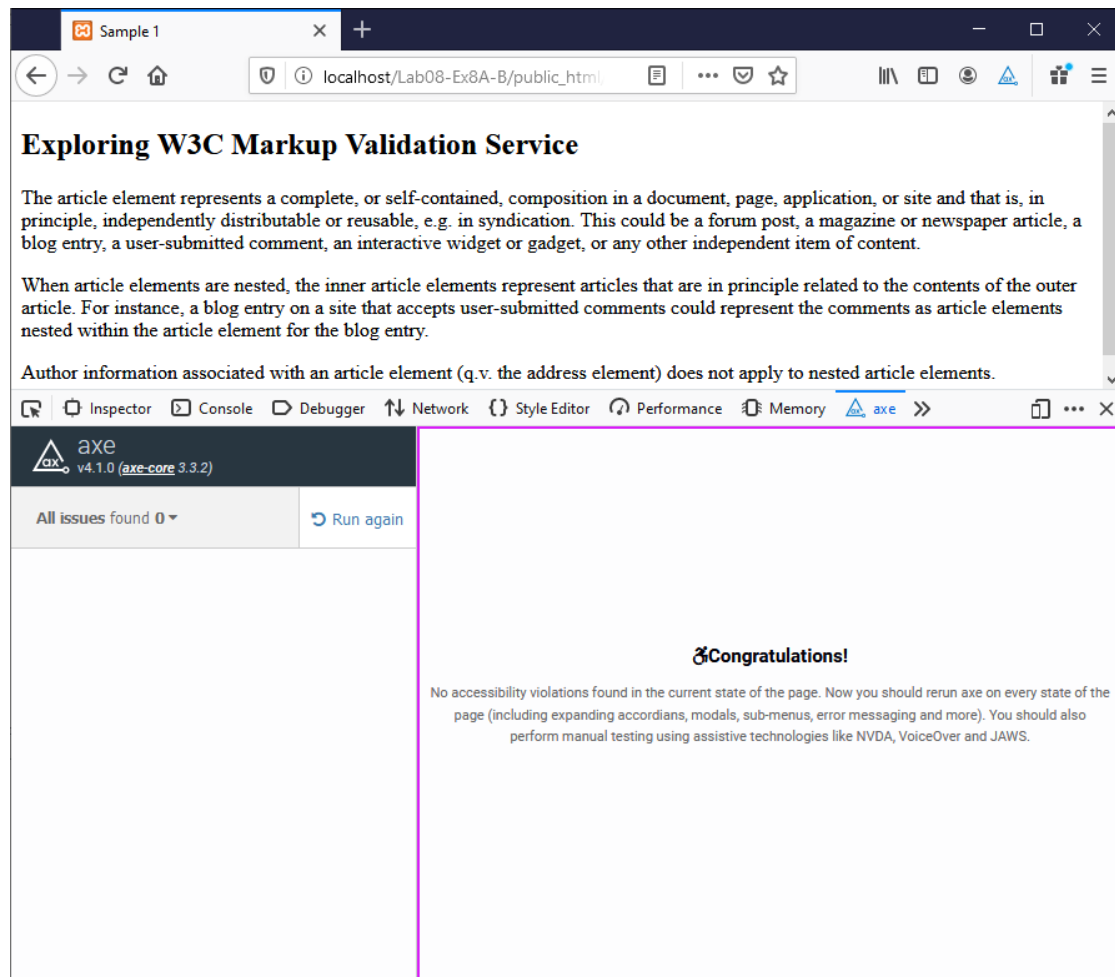
Issue	Count
Document must have one main landmark	1
All page content must be contained by landmarks	1

The first issue, "Document must have one main landmark", is selected. Its details are as follows:

- Issue description:** Ensures the document has only one main landmark and each iframe in the page has at most one main landmark. Impact: moderate. [Learn more](#)
- Element location:** `html`
- Element source:** `<html lang="en">`
- To solve this violation, you need to:**
 - Fix the following: Document does not have a main landmark

The issue tags are: category: semantics, best-practice.

- h. Edit the file to add the `<main>` tag, then run the accessibility test again to ensure all issues are resolved:



- 8.3 Repeat the above steps with the remaining files, 'sample2.html' and 'sample3.html', and fix all issues found.

Tip: if you get stuck, you can click the 'Learn more' link that's provided by axe for each issue.

- 8.4 Once you've fixed all validation and accessibility issues, submit the three files for your practical assignment.

9. SUBMISSION OF LAB ASSIGNMENT

- 9.1 In order to receive credit for this Lab assignment, you must submit your completed work to xSiTe LMS before the end of the week. To submit your work:
- Save all files and close NetBeans.
 - In File Explorer, navigate to the location where you saved your project and right-click on the folder name, then select 'Send to -> Compressed (zipped) folder' and ZIP up your entire NetBeans project. **Note: only .zip format is acceptable, do not use .rar, .7z, or any other format.**
 - In the ICT1004 module on xSiTe, go to **Assessments->DropBox** and locate the Dropbox folder corresponding to this Lab. Click the link to open the Dropbox then

hit the **Add a File** button to submit your .zip file. You may also add comments if desired. Be sure to hit **Submit** to complete your submission.