ATTAFI YOUSRA

Selenium Certified Tester

Part 2
Web Technologies

Knowledge levels of learning objectives

01

K1: remember

02

K2: understand

03

K3: apply

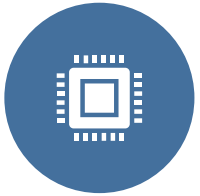
04

K4: analyze

Understanding HTML and XML

Objectives of the chapter

STF-2.1 (K3) UNDERSTAND AND WRITE HTML AND XML DOCUMENTS

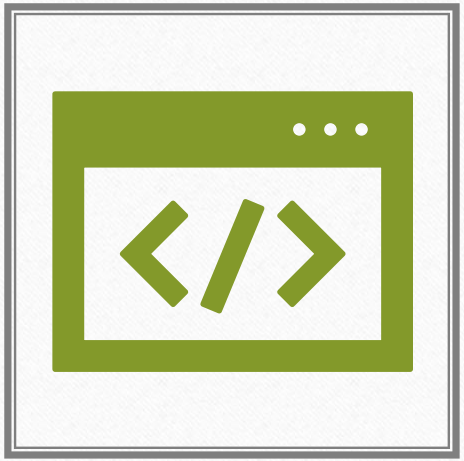


STF-2.2 (K3) APPLY XPATH TO SEARCH FOR XML DOCUMENTS

STF-2.3 (K3) APPLY CSS LOCATORS TO FIND HTML DOCUMENT ELEMENTS

The HTML

- Essentially, HTML semantically describes the structure of a Web page.
- An HTML document is a simple text file containing elements that specify certain contextual meanings . The elements combine to determine how a browser should display these parts of the document.
- The HTML language is universally applicable. When written correctly, any browser on any computer system can display the page correctly.



The HTML

- For a long time, the HTML version used was stable with version 4.01. But in 2014, the governing body that controls HTML released the current version, HTML 5.

- HTML is a flexible language, allowing some variation in how tags are used.

- XML is a more restrictive language than HTML, requiring that each page be "well formed" , each opening tag being balanced by a closing tag. When written in a well-formed way, HTML is a subset of XML.

HTML/ syntax

- HTML elements are inserted often surrounded by chevron-defined tags.

- Some tags introduce content directly into the page being viewed (for example, will result in an image being placed on the page.)

- Other tags surround and provide semantic information about how the element is to be rendered (as seen above for the header element <h1>....</h1>).

HTML/ syntax

- HTML elements usually have a start tag and an end tag. The end tag is the same as the start tag except that the end tag is preceded by a slash as shown below:

** Text paragraph **

- Some elements can be closed in the opening tag; for example, the empty line break element as follows:

**
**

- A less strict implementation of the line break, which could cause problems for some browsers would be:

**
**

<! DOCTYPE>	Defines document type (not required in HTML 5)
<html> … </html>	Indicates the root of the HTML document
 	Inserts a single line break
<head> … </head>	Contains the page header information
<body> … </body>	Defines the main content of the document
…	Defines a paragraph
…	Defines a section in the document
<!-- … -->	Defines a comment

Basic HTML tags Structure

List and Table Tags

Beacon	Use
`<ul> … </ul>`	Defines an unordered list (bulleted)
`<ol> … </ol>`	Defines an ordered (numbered) list
`<li> … </li>`	Defines a list item (for or)
`<table> … </table>`	Defines an HTML table
`<tr> … </tr>`	Defines a table line
`<th> … </th>`	Defines the column heading of a table
`<td> … </td>`	Defines a table data cell
`<thead> … </thead>`	Defines an HTML table header
`<tfoot> … </tfoot>`	Defines an HTML table footer

Forms

Beacon	Use
<form> … </form>	Defines an HTML form for user input
<input>	Defines an input control. The control type is defined by the type of attribute type=. Possible types include text, radio button, check box, form submission, etc.
<textarea> … </textarea>	Defines a multiple line input command
<button>	Defines a clickable button
<select> … </select>	Defines a drop-down list. When used with <option>.... </option> tags, the author can set a drop-down list and fill it in
<fieldset> … </fieldset>	These tags allow the author to group linked elements in a form. When used with the <legend> tag, it displays a box named around the controls.

The XML

•XML (eXtensible Markup Language) is a markup language used to define document formatting rules in a machine and human readable manner. It was designed to:

- Increase simplicity and ease of use on the Web.
- Enable the representation of data structures that can be created on the fly.
- Display data with emphasis on presentation.
- Be software and hardware independent
- Transmit and store data in a readable format.

•XML tags are not predefined as are HTML tags. Instead, they are specified by the creator of the XML document.

•XML separates data from how it is presented.

XML: Tags and elements

- Each opening tag, <*TAG*>, has a corresponding closing tag,
- </*TAG*>. The complete construction is called an element.
- Elements may be incorporated into other elements.
- An XML document always forms a tree structure. The first element of usually indicates that it is an XML document.

XML: Attributes

- In addition to tags, XML supports attributes that provide additional information about the element they are associated with. An attribute consists of a pair of terms separated by an equal sign. For example:
- <person gender=”female”>
- The attribute is contained in the element tags. Rather than using an attribute, the same information can be used as an element.

XML: Attribute to element

- Attributes are not as flexible as elements. For example, the following points are underlined by the W3C, the body that controls the XML standard:

  - Attributes cannot contain multiple values (elements can).
  - Attributes cannot contain tree structures (elements can).
  - The attributes of are not easily expandable (for future changes).

The XML tree structure

- An XML document is composed of **nodes**. In practice, a node can be:
  - The document itself, represented by its **root node** ;
  - A processing instruction namespace , this is most often the <?xml version… ? > node
  - Comments;
  - XML elements;
  - XML attributes;
  - The textual content of the elements.

The XML tree: Relationships between nodes

- The following relationships are defined:
  - Parent relationship: all elements and attributes have a parent. In our example, sailor is parent of name and first name.
  - Child relationship: A knot can have as many children as you want. In our example, sailor has two children: name and first name, the name node has no child.
  - Brother relationship (sibling): two knots are brothers if they have even father. In our example, name and first name are brothers.
  - Ancestor: the ancestors of a knot are their father, their father, etc… up to the **root node**, ancestor of all nodes in the XML document.
  - Relation des descendents: sons, sons of sons, etc. The root node of the document has as offspring all the nodes of the XML document.

Understand XPATH

XPath

XPath is an XML file query language.

It uses path expressions to identify specific elements and navigate to an XML document.

Since HTML is a subset of XML, XPath can also be used to search HTML documents.

XPath expressions to select nodes

Expression	Description
The Node	Selects all items with the name "The Node"
/	Selects from the root element
//	Returns the descendants of the current element
.	Returns the current element
..	Selects the parent of the current element
@	Selects an attribute of the current element

Xpath: the predicates

- Predicates are used to find a specific element or an element that contains a specific value. Predicates are always surrounded by brackets and appear directly after the name of the element.

Examples of predicates

Path expression	Result
`/marine/marine[1]`	The first element “marine”
`/marine/marin[last()]`	The last element “marine”
`//note[@lang]`	All remark elements with the attribute "lang"
`//marine[@id=13]`	All “marine” elements whose “id” attribute is 13

XPath wildcards expressions

Beacon	Use
*	Matches any element of the node
@*	Matches any attribute of the node
Node()	Matches any node of any type

XPath operators

Operator	Description	Example
|	Several paths	//Magazine | //CD
+, -	Addition, Subtraction	2 + 2 2-2
*	Multiplication	8 * 7
div	Division	14 Div 2
mod	Modulo (the rest after division)	mod 7 mod 3
=, !=, <, >, >=, <=	Comparison operators	price=4.35
!=	Different	price!= 4.35
or, and	logical operators	lang="en"
||	Concatenation	"en" || "glish"

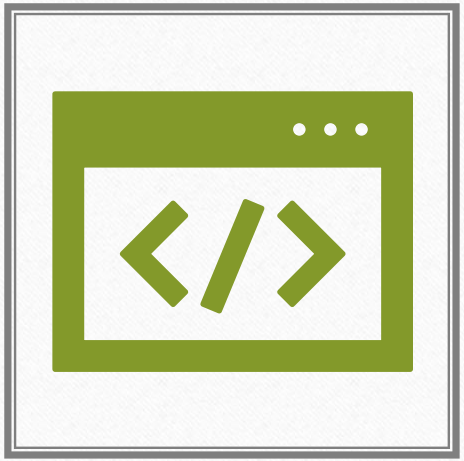
XPath chain handling functions

Operator	Description
string(arg)	Returns the string value of the argument
substring(str, start, len)	Returns a subset of a chain of length "len" from "start"
string-length(str)	Returns the length of the string. If there is no argument passed, return the running node length
compare(str1, str2)	Returns -1 if str1 < str2, 0 if strings are equal, +1 if str1 > Str2
concat(str1, str2, …)	Returns a concatenation of all strings of typed characters
upper-case(str)	Converts string to upper case
lower-case(str)	Converts string to lowercase
contains(str1, str2)	Returns TRUE if str1 contains str2
starts-with(str1, str2)	Returns TRUE if str1 starts with str2
ends-with(str1, str2)	Returns TRUE if str1 ends with str2

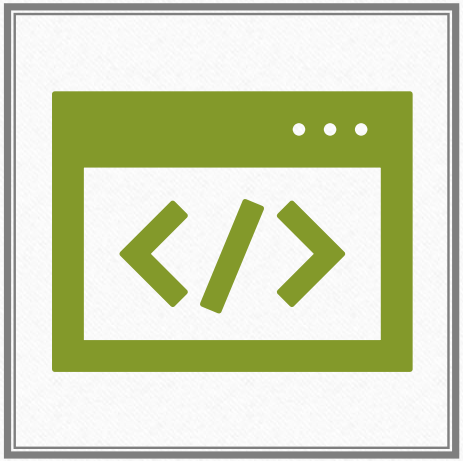
Understanding CSS selection

The CSS

- CSS stands **for Cascading Style Sheetset is primarily used to determine** how the different HTML elements of a document set are to be rendered to the screen, on paper or on other media. External CSS style sheets are stored in CSS files.
- When applied to Selenium tests, CSS is particularly useful for finding HTML elements, allowing you to automate tests on the browser.



CSS: use



- CSS can be used in three different forms in HTML documents:
  - An external style sheet : Each HTML page must include a reference to the external style sheet file within the <link>element that goes into the <head> section.
  - An internal style sheet: when an HTML page must have a particular style ; Styles are defined in the <style> element within the < head> section of the document.
  - An online style: applies to a specific element and is added directly to the element as an attribute.

CSS: use

- When multiple CSS styles are defined for the same element, the value of the last read style sheet will be used. Therefore, the order in which a style will be used is defined (starting at the top) as follows:
  - Online style (as an attribute in an HTML element).
  - External and internal style sheets defined in the header section.
  - The default browser value.

CSS and Selenium

- CSS selectors are used to locate elements in the HTML document based on the element name, ID, class, attribute, or other specifiers.

- In many cases, XPath can be used in the same way to find certain elements.

XPath vs CSS

CSS	Xpath	Result
div.even	//div[@class=”even”]	Div elements with a class="even" attribute
#login	//*[@id=”login”]	An element with id="login"
*	//*	All elements
input	//input	All input elements
div input	//div//input	All input elements within all div elements
div > input	//div/input	All input elements that have the div element as parent
br + p		All elements p that are placed immediately after element br
p ~ br		all elements p that are placed immediately before item b

CSS selection by attributes

[lang]	All elements with lang attribute
[lang=en]	All elements with the lang attribute taking the value in
[lang =en]	All elements with the lang attribute starting with the string in
[lang | =en]	All elements that have the lang attribute equal to or starting with the string followed by a hyphen
[lang$=en]	All items that have the lang attribute ending in the string in
[lang~=en]	All elements that have the lang attribute whose value is a list of words separated by spaces, one of which is exactly the string in
[lang*=en]	All elements that have the lang attribute containing the string in

CSS selection by form elements

:checked	Selects all checked items (for check boxes, radio buttons and options that are set to enabled)
:default	Selects any form item that defaults from a group of related items
:defined	Selects all items that have been defined
:disabled	Selects all items that have been disabled
:enabled	Selects all items that have been enabled
:focus	Selects the element that is the focus
:invalid	Selects the form elements that are not validated
:optional	Selects form items that do not have the 'required' attribute enabled
:out-of-range	Selects all input elements whose current value is outside the min to max value area
:read-only	Selects items that are not user editable
:read-write	Selects items that can be edited by the user
:required	Selects form elements where the 'required' attribute is enabled
:valid	Selects form items that are successfully validated
:visited	Selects links a user has already visited

CSS Selector: **Others**

:not(<selector>)	Selects items that do not match the specified selector
:first-child	Selects all items that are the first children of their parent elements.
:last-child	Selects all items that are the last children of their parent elements
:nth-child(<n>)	Selects all elements that are the nth children of their parent elements
:nth-last-child(<n>)	Selects all items that are nth children of their parent elements, counting from the last child
div:nth-of-type(<n>)	Selects all elements that are nth div children of their parent elements

Conclusion of the chapter

To excel with Selenium, it is crucial to grasp HTML, CSS, XML, and XPath. However, the information provided here serves only as an introductory overview of these topics.

The W3C (World-Wide-Web Consortium) is a global community that works in collaboration with diverse organizations and the public to establish and advance Web standards, encompassing XML and XPath.

Thank You

ATTAFI YOUSRA
EXPERT EN TEST LOGICIEL
@ : ATTAFI.YOSRA@HOTMAIL.FR
LINKEDIN : Yousra ATTAFI