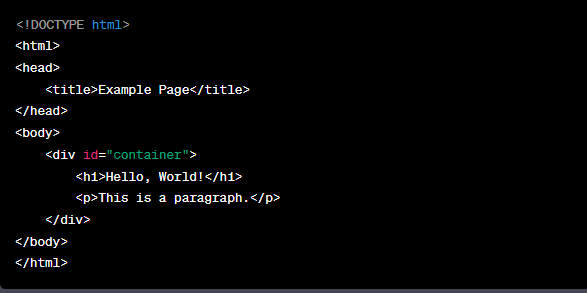
2.

2.1. Absolute vs. Relative XPath

2.1.1 Absolute XPath : Absolute XPath is an XPath expression that specifies the complete path to the desired element from the root of the document.

Example :

Consider the following HTML snippet:



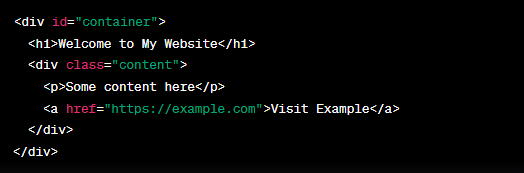
Let’s say you want to select the `<p>` element using an absolute XPath expression.

The absolute XPath for the `<p>` element would be: `/html/body/div/p`.

2.1.2 Relative XPath Relative XPath is a way to locate elements on a web page by specifying a path that starts from a particular element rather than from the root of the document.

Here's an explanation of relative XPath with examples:

Consider the following HTML structure:



1. Locating the "Welcome to My Website" Heading:

XPath : //div[@id='container']/h1 // highlight this

1. Locating the "Some content here" Paragraph:

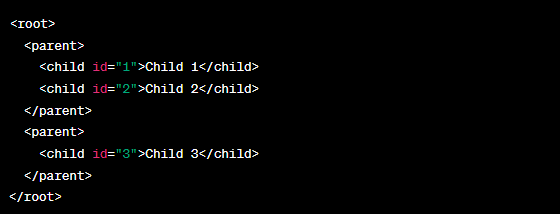
XPath : //div[@class='content']/p

1. Locating the "Visit Example" Link:

XPath : //div[@class='content']/a[text()='Visit Example']

2.2 XPath Axes :

XPath axes are a set of methods that allow you to traverse the hierarchical structure of an XML or HTML document to locate elements based on their relationships with other elements. There are several axes available in XPath, and they provide different ways to navigate through the document tree. Here are the main XPath axes explained with examples:



1. Ancestor Axis (ancestor::) :

XPath : //child[@id='3']/ancestor::parent

1. Parent Axis (parent::) :

XPath : //child[@id='1']/parent::parent

1. Child Axis (child::) :

XPath : //parent/child

1. Descendant Axis (descendant::) :

XPath : //parent[@id='1']/descendant::child

1. Following Axis (following::) :

XPath : //parent[@id='1']/following::child

1. Preceding Axis (preceding::) :

XPath : //child[@id='2']/preceding::child

1. Following-Sibling Axis (following-sibling::) :

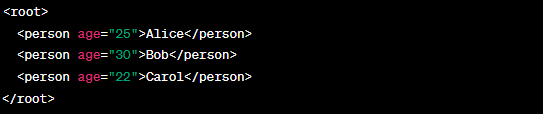
XPath : //child[@id='1']/following-sibling::child

1. Preceding-Sibling Axis (preceding-sibling::) :

XPath : //child[@id='3']/preceding-sibling::child

2.3 XPath Predicates :

XPath predicates are used to filter elements or nodes in an XML or HTML document based on certain conditions. Predicates are added to XPath expressions within square brackets [...], and they allow you to select only the nodes that satisfy the given condition.



1. Selecting Elements with Specific Attribute Values :

XPath : //person[@age='30']

1. Selecting Elements with Numeric Attributes :

XPath : //person[number(@age) > 25]

1. Selecting Elements with Text Content :

XPath : //person[text()='Alice']

1. Selecting Nth Element :

XPath : //person[2]

2.4 Common XPath Functions:

XPath functions enhance your ability to navigate and manipulate the document's structure and content.

Here are some common XPath functions and their usage:

1. text() Function:

This function returns the text content of the selected node.

XPath : //p/text()

1. @attribute\_name:

This function selects the value of the specified attribute.

XPath : //a/@href

1. count():

This function counts the number of nodes that match a given XPath expression.

XPath : count(//p)

1. concat():

This function concatenates two or more strings.

XPath : concat("Hello, ", "world!")

1. contains():

This function checks if a given string contains a specific substring.

XPath : //div[contains(@class, "important")]

1. starts-with():

This function checks if a given string starts with a specific substring.

XPath : //img[starts-with(@src, "https://example.com")]

1. position():

This function returns the position of the current node in its parent's list of child nodes.

XPath : //li[position() = 2]

1. last():

This function returns the position of the last node in the context.

XPath : //tr[last()]

1. sum():

This function calculates the sum of numeric values in a selected set of nodes.

XPath : sum(//price)

1. normalize-space():

This function removes leading and trailing whitespace and collapses internal whitespace into a single space.

XPath : normalize-space(//p)