

RELATIVE

XPath pattern based on the attribute / text

ATTRIBUTE

//tagName[@attribute='value']

TEXT

//tagName[text()='value']

PARTIAL ATTRIBUTE

//tagName[contains(@attribute,'value')]

PARTIAL TEXT

//tagName[contains(text(),'value')]

PARENT - CHILD

XPath pattern based on the (grand) parent to (grand) child relationships

PARENT TO CHILD

(xpath of parent)/child-tagName

CHILD TO PARENT

(xpath of child)/parent::parent-tagName

CHILD TO GRAND PARENT

(xpath of child)/ancestor::ancestor-tagName

GRAND PARENT TO CHILD

(xpath of ancestor)//child-tagName

LAST CHILD

(xpath of parent)/child-tagName[last()]

COLLECTION

XPath pattern based on the indexes with(out) relations.

Note: index starts at 1

USING ITEM INDEX

(xpath)[index]

USING CHILD ITEM INDEX

(parent-xpath)/child-tagName[index]

USING AND CONDITION

//tagName[@att1='v1' and @att2='v2']

USING OR CONDITION

//tagName[@att1='v1' or @att2='v2']



Use Ruto, the opensource xpath finder if you find difficult to write your own xpaths.

It is available as Chrome, Firefox, Edge extensions

AXES

XPath pattern based on the sibling & Cousin relationships

YOUNGER SIBLING

(xpath)/following-sibling::sibling-tagName

ELDER SIBLING

(xpath)/preceding-sibling::sibling-tagName

YOUNGER COUSIN

(xpath)/following::cousin-tagName

ELDER COUSIN

(xpath)/preceding::cousin-tagName



www.testleaf.com

XPATH CHEATSHEET