|  |  |
| --- | --- |
|  |  |
| **What is CSS Selector/locators?** |  |
| * CSS (Cascading Style Sheets) is defined to display HTML in structured and colourful styles are applied to webpage. * Selectors are patterns that match desired element against elements in a tree |  |
| **Why to use CSS Selector/locators?** |  |
| * It is one of several technologies that can be used to select nodes in an XML document. * When we don't have an option to choose Id or Name, we should prefer using CSS selectors/locators as the best alternative.   CSS has more Advantage than Xpath   1. CSS is much faster and simpler than the Xpath. 2. In IE Xpath works very slow, whereas Css works faster when compared to Xpath. |  |
| **What is Standard Syntax of CSS Locator?** |  |
| Syntax: tasv  tag[attribute<special character>=’value’]   |  | | --- | | Alternative Symbols of id:   * “#” -> Id | | Alternative Symbols of class:   * “.” -> Class | | Using special character/wildcards in CSS Selectors:   1. “^” -> Represents the starting text 2. “$” -> Represents the ending text 3. “\*” -> Represents the text contained | | Finding Children: using ‘>’ symbol |   Note: Attribute can be any attribute, not limited to class/id, those are just more popular |  |
| **How to use CSS locator in short?** |  |
| |  | | --- | | Alternative Symbols of id:   * “#” -> Id * Example: input[id=displayed-text] can be written as  1. input#displayed-text 2. #displayed-text | | Alternative Symbols of class:   * “.” -> Class * Example: input[class=displayed-class] can be written as  1. input.displayed-class 2. .displayed-class  * For Nodes which contains multiple class avlues, We can write them as : .class1.class2.class3 -> Until we find a unique element * .inputs.displayed-class | | Using special character/wildcards in CSS Selectors:   1. “^” -> Represents the starting text 2. “$” -> Represents the ending text 3. “\*” -> Represents the text contained   Examples:  input[class^='inputs'] -> Two matching nodes  input[class$='class'] -> One matching node  input[class='displayed-class'] - No matching nodes  input[class\*='displayed-class'] -> One matching node | | Finding Children: using ‘>’ symbol  Examples:  fieldset>table  fieldset>#product -> One matching node  fieldset>button -> One matching node  fieldset>a  fieldset>input#name | |  |
| **How to use CSS locator in detail?** |  |
| |  | | --- | | Details of Symbols: | | **Alternative Symbols of id**:   * “#” -> Id * Example: input[id=displayed-text] can be written as  1. input#displayed-text 2. #displayed-text | | **Alternative Symbols of class**:   * “.” -> Class * Example: input[class=displayed-class] can be written as      1. input.displayed-class 2. .displayed-class  * For Nodes which contains multiple class avlues, We can write them as: .class1.class2.class3 -> Until we find a unique element * .inputs.displayed-class      * .blocks-page.blocks-page-rich\_text | | Using special character/wildcards in CSS Selectors:   1. “^” -> Represents the starting text 2. “$” -> Represents the ending text 3. “\*” -> Represents the text contained   Examples:    input[class^='inputs'] -> Two matching nodes  input[class$='class'] -> One matching node    input[class='displayed-class'] - No matching nodes  input[class\*='displayed-class'] -> One matching node | | Finding Children: using ‘>’ symbol.  Examples:    fieldset>table  fieldset>#product -> One matching node    fieldset>button -> One matching node    fieldset>a    fieldset>input#name | |  |