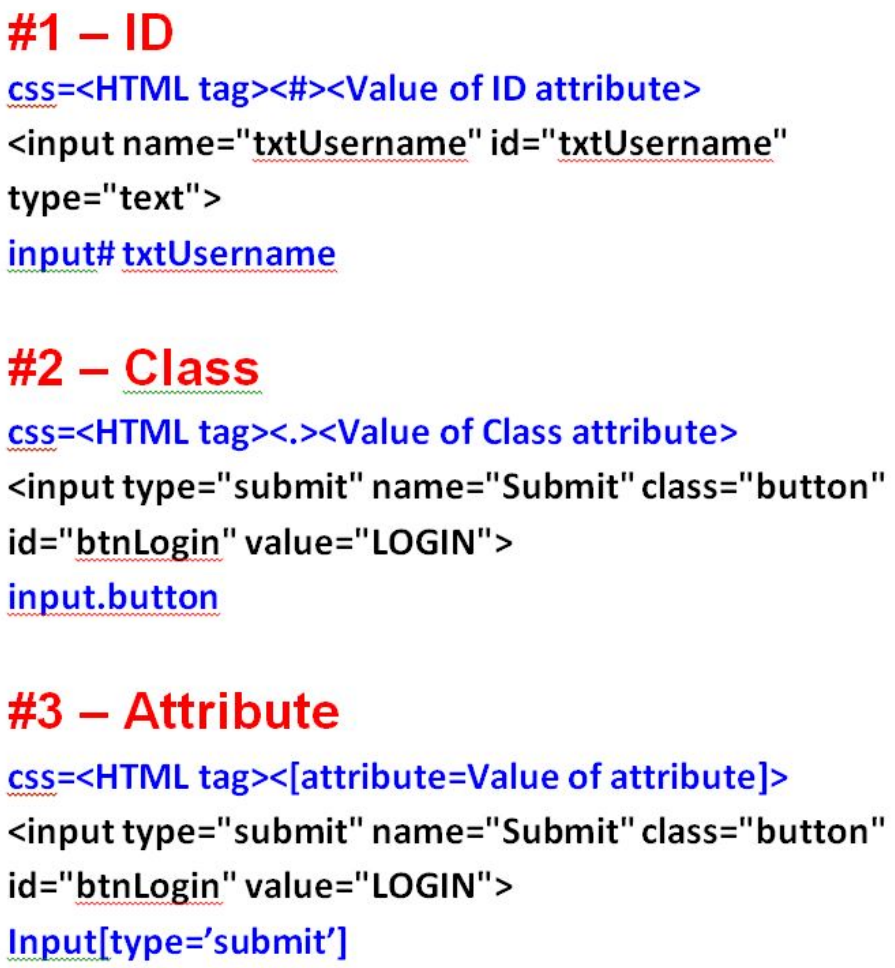
Selenium Sudhakar

Java SQL Agile_Scrum CUCUMBER GuideWIre

Browser	Selenium Grid	Day_013_InvocationCount	TestNG Annotations	Day_018_Waits
Get_vs_Navigate	Selenium Limitations	Day_005_TestNG_DataProvider_Lab1	TestNG_Parameters1	Day_019_Frames
Day_004_Dropdown	Selenium Featues	Day_006_TestNG_DataProvider_Lab2	Day_014_015_Parallel_Methods_Tests	Day_020_Multiple_Windows
OR1_applicationProperties	Selenium Challanges	Day_007_Browser_Compatability_Testing	Day_016_ParallelClasses	Day_021_Java_Script_Executer
OR1_Excel	Web Table	Day_025_ExcelUtil_Apache_POI	Day_017_Multiple_Suites	Day_022_Action_class
Xpath_vs_cssSelector	Capcha	Proxy Setting	Day_032_TestNG_Groups	Day_027_Alerts
cssSelector	Assertions	Dimesions	Day_034_TestNG_Listeners	Day_026_TakeScreenShot
Day_024_Advanced_Xpaths	Page Object Model	Keyword Driven	TestNG_Priority	Day_023_Robot_Class
Day_41_Exceptions	Page Factory Model	Interfaces	dependsOnMethods	Day_025_ExcelUtil_Apache_POI
Web Element Border	Web Element Color	Classes in Selenium	BrokenLinks	Emailable Report
Locators in Selenium	FindElement vs Find Elements	Number of Hyperlinks	Extent Report1	SSL Certificate
Page Load Strategy	Relative Locators	Selenium Suite	Chrome Options	Preserve Order
Maven Life Cycle - Pom.xml	HeadLess Browser	is Displayed - is Enabled - is Selected	Navigate_Get	Quit_Close
Tell me about your self	HeadLess Browser	is Displayed - is Enabled - is Selected	Navigate_Get	Quit_Close

CSS Selectopr best youtube video session https://www.youtube.com/watch?v=VYIDZ57mMy4



#1 – ID

css=<HTML tag><#><Value of ID attribute>

<input name="txtUsername" id="txtUsername"
type="text">

input# txtUsername

#2 – Class

css=<HTML tag><.><Value of Class attribute>

<input type="submit" name="Submit" class="button"
id="btnLogin" value="LOGIN">

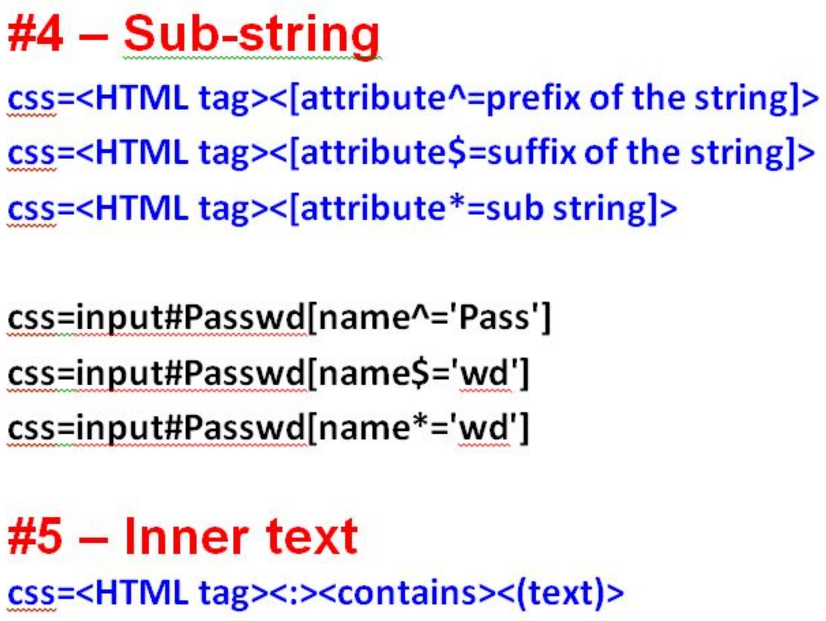
input.button

#3 – Attribute

css=<HTML tag><[attribute=Value of attribute]>

<input type="submit" name="Submit" class="button"
id="btnLogin" value="LOGIN">

Input[type=’submit’]



#4 – Sub-string

css=<HTML tag><[attribute^=prefix of the string]>

css=<HTML tag><[attribute$=suffix of the string]>

css=<HTML tag><[attribute*=sub string]>

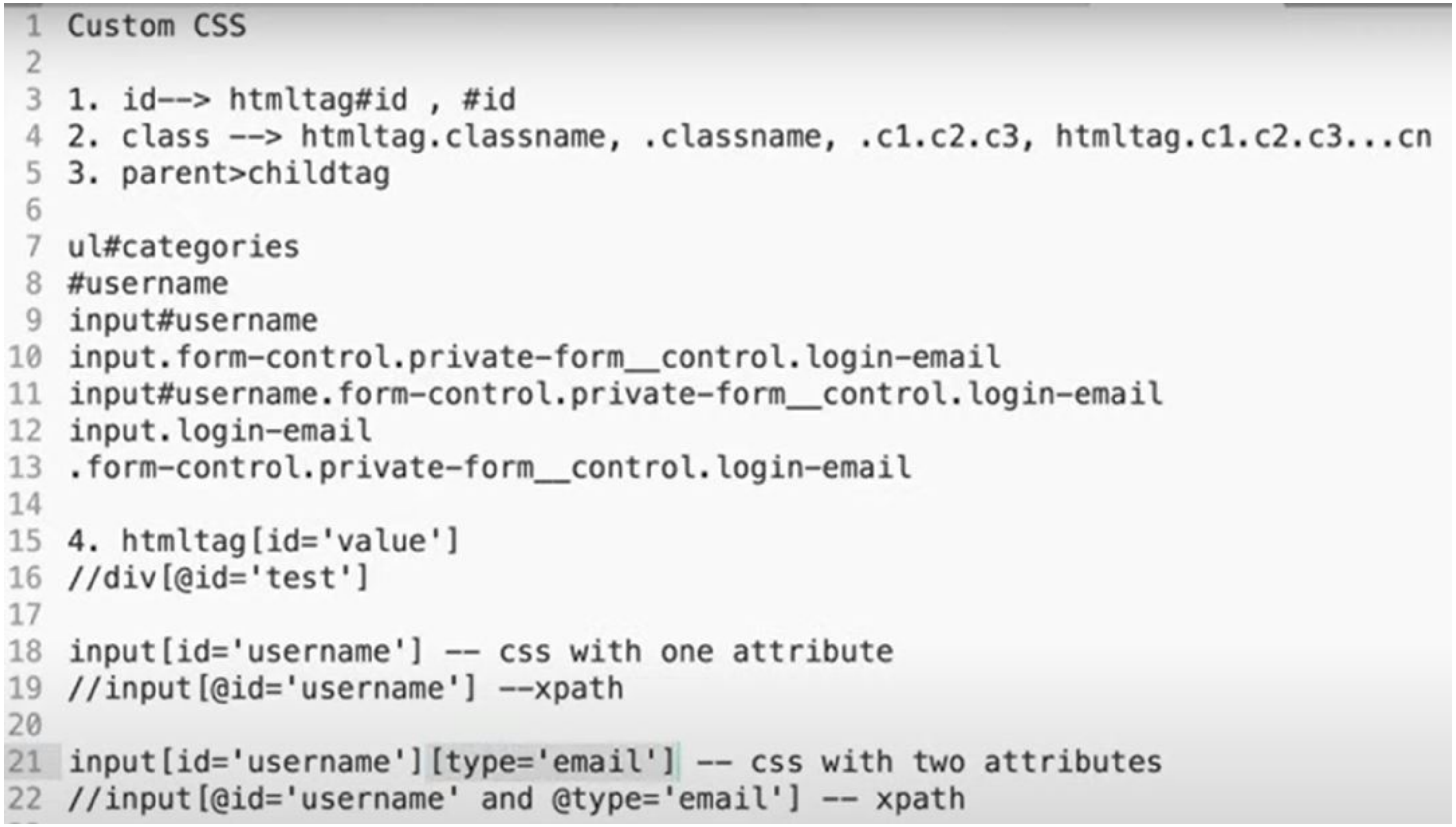
css=input#Passwd[name^='Pass']

css=input#Passwd[name$='wd']

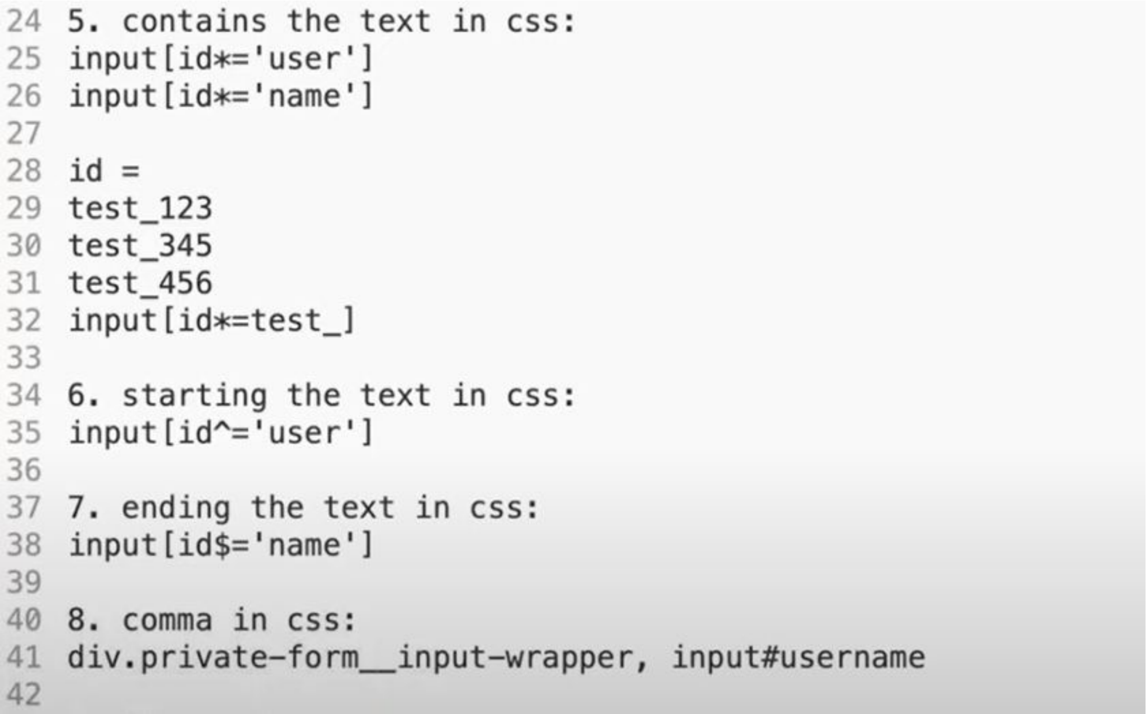
css=input#Passwd[name*='wd']

#5 – Inner text

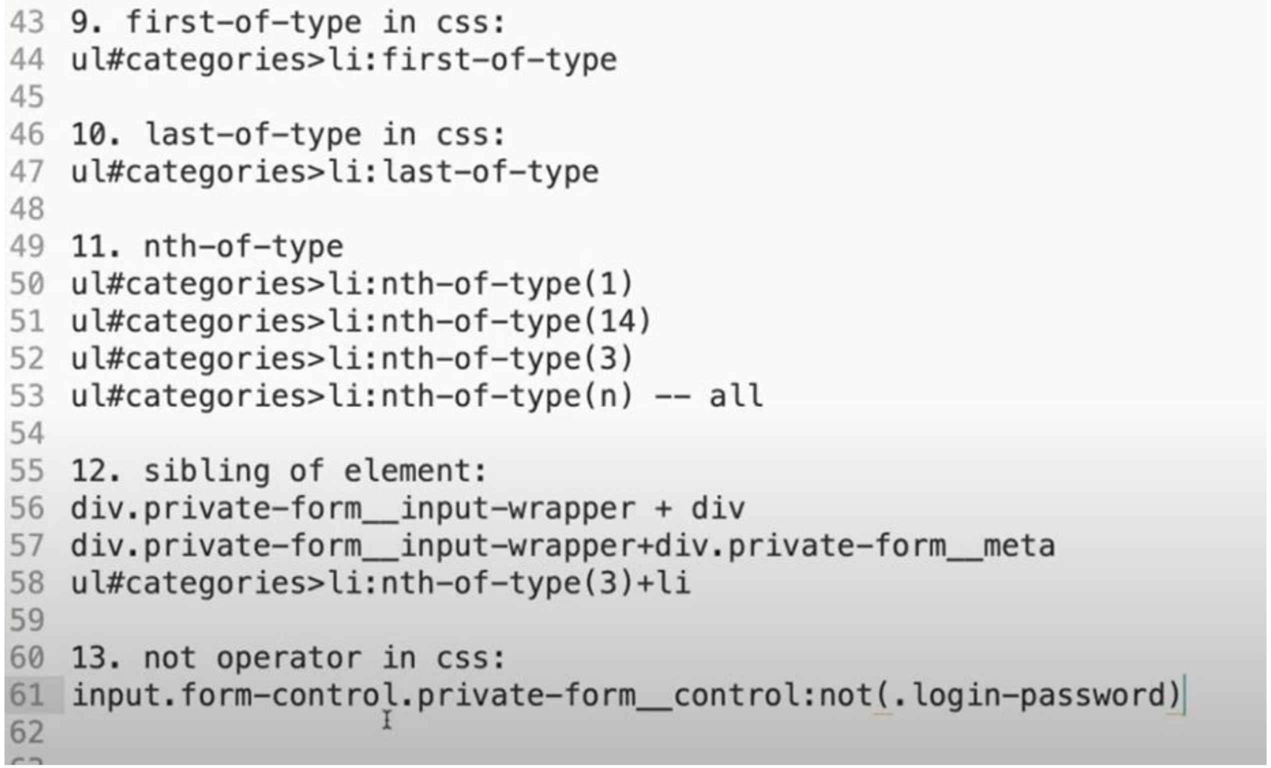
css=<HTML tag><:><contains><(text)>



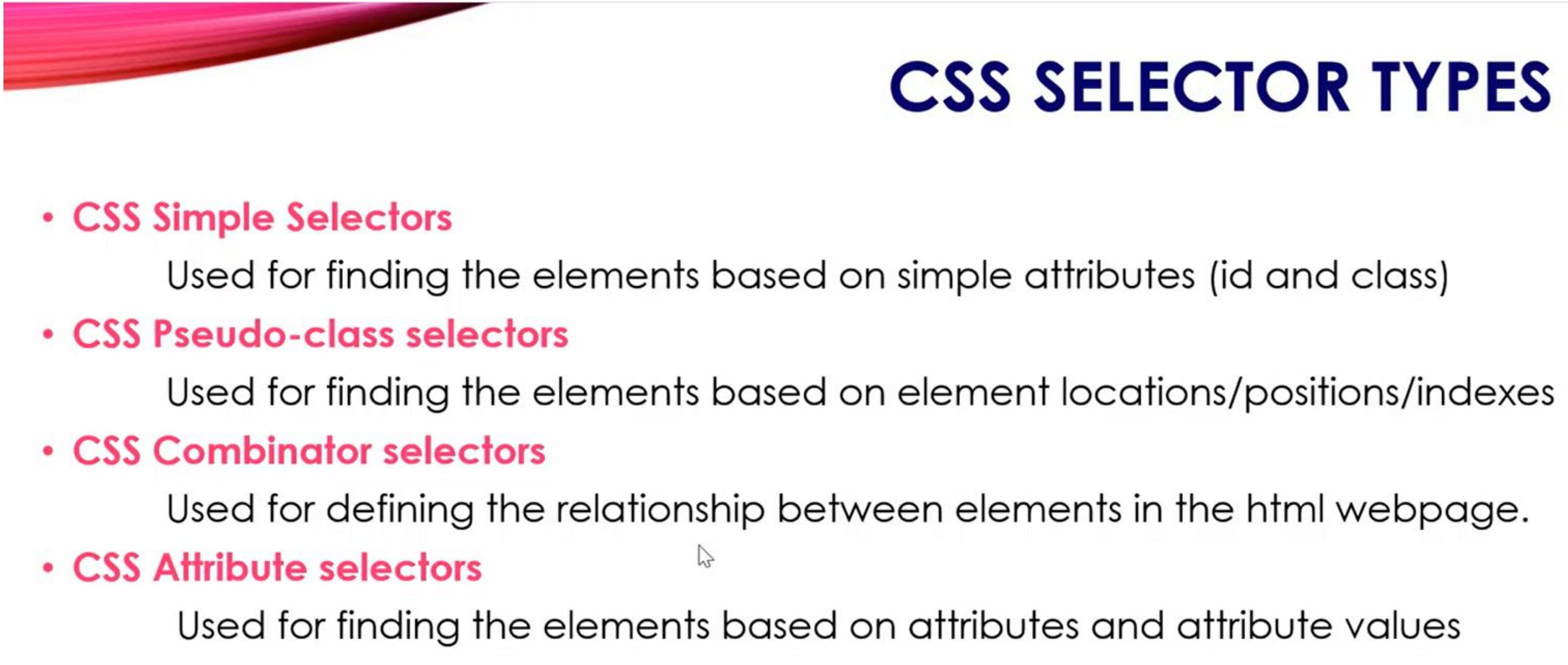
```
Custom CSS

1. id--> htmltag#id , #id
2. class --> htmltag.classname, .classname, .c1.c2.c3, htmltag.c1.c2.c3...cn
3. parent>childtag

ul#categories
#username
input#username
input.form-control.private-form__control.login-email
input#username.form-control.private-form__control.login-email
input.login-email
.form-control.private-form__control.login-email

4. htmltag[id='value']
//div[@id='test']

input[id='username'] -- css with one attribute
//input[@id='username'] --xpath

input[id='username'][type='email'] -- css with two attributes
//input[@id='username' and @type='email'] -- xpath
```



```
5. contains the text in css:
input[id*='user']
input[id*='name']

id =
test_123
test_345
test_456
input[id*=test_]

6. starting the text in css:
input[id^='user']

7. ending the text in css:
input[id$='name']

8. comma in css:
div.private-form__input-wrapper, input#username

```



```
9. first-of-type in css:
ul#categories>li:first-of-type

10. last-of-type in css:
ul#categories>li:last-of-type

11. nth-of-type
ul#categories>li:nth-of-type(1)
ul#categories>li:nth-of-type(14)
ul#categories>li:nth-of-type(3)
ul#categories>li:nth-of-type(n) -- all

12. sibling of element:
div.private-form__input-wrapper + div
div.private-form__input-wrapper+div.private-form__meta
ul#categories>li:nth-of-type(3)+li

13. not operator in css:
input.form-control.private-form__control:not(.login-password)
```



CSS SELECTOR TYPES

- **CSS Simple Selectors**

  Used for finding the elements based on simple attributes (id and class)

- **CSS Pseudo-class selectors**

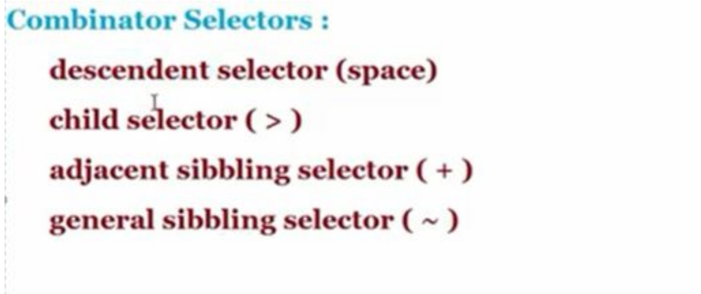
  Used for finding the elements based on element locations/positions/indexes

- **CSS Combinator selectors**

  Used for defining the relationship between elements in the html webpage.

- **CSS Attribute selectors**

  Used for finding the elements based on attributes and attribute values



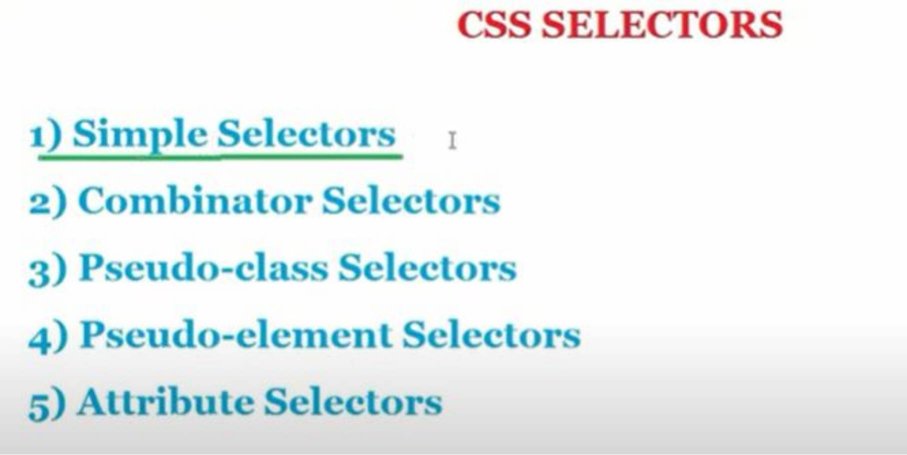
Combinator Selectors :

descendent selector (space)

child selector (>)

adjacent sibbling selector (+)

general sibbling selector (~)



CSS SELECTORS

1) Simple Selectors

2) Combinator Selectors

3) Pseudo-class Selectors

4) Pseudo-element Selectors

5) Attribute Selectors

Summary

Syntax for Locator Usage

Method	Target Syntax	Example
By ID	id= *id_of_the_element*	id=email
By Name	name=*name_of_the_element*	name=userName
By Name Using Filters	name=*name_of_the_element* *filter*=*value_of_filter*	name=tripType value=oneway
By Link Text	link=*link_text*	link=REGISTER
Tag and ID	css=*tag*#*id*	css=input#email
Tag and Class	css=*tag.class*	css=input.inputtext
Tag and Attribute	css=*tag*[*attribute*=*value*]	css=input[name=lastName]
Tag, Class, and Attribute	css=*tag.class*[*attribute*=*value*]	css=input.inputtext[tabindex=1]