CSS, XPATH and Playwright Specific

CSS Selectors

1) id locator
locator(email)

2) class locator
locator(email)

3) name locator
locator(tagname[name="....."])

Ex:
<input type="text" name="email" placeholder="email" />
<input type="password" name="pass" placeholder="pass" />
tagname => input
atrribute_name = attribute_value ===> attribute

4) attribute locator
locator([name="email"]) ==> email webelement
locator([type="password"]) ==> password webelement

5) tag name
locator(tagname) ==> locator(input)

6) multiple attributes

locator([attr_name = 'attr_value'][attr_name = 'attr_value'])

7) tagname with id
locator(tagnameid)
==> locator(input).nth(number) ==> email web element
==> locator(input#email) ==> email web element

8) tagname with class
locator(tagnameclass)

9) direct child selector(>)

10) descendant selector(space)

The above 2 selector will target child using the parent

```
<div class="card">
    <h2>Product Title</h2>
    <button>Buy</button>

    <div class="footer">
        <button>Add</button>
    </div>
</div>
```

Problems with CSS selectors(some places where CSS selectors wont work):-

1) We cannot use CSS to identify any web element using the text

```
<button>Buy</button>
<button>Add</button>
<h1>Welcome</h1>
```

locator("text='Welcome'")

playwright specific

2) When we want to navigate from child to parent in CSS

```
<div class="footer">
    Enter Email:<input type="email">
</div>
```

locator(.footer > input).fill("....")

I want to target the parent starting from child

```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

Write the logic to click on the samsung s25 products buy button?

locator(button).nth(1) ==> Buy

```
<div class="product">
    <h2>Samsumg s25</h2>
    <button>Buy</button>
</div>
```

Solution is XPATH

1) text() = "Add" ==> Add button
<h1>Welcome</h1>
text() = "Welcome"

2)

```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

text() = 'Samsung s25'/../text() = 'Buy'

```
<div class="product">
    <h2>Samsumg s25</h2>
    <button>Buy</button>
</div>
```

3) We cannot use logical operators in CSS
and, or

<input type = "text" name = "email" placeholder = "email" />

locator(type="text" and name="email")
locator(type="text" or name="email")

==========
XPATH ==> locator(type="text" and name="email")
locator(type="text" or name="email")

4) We cannot use CSS selector to navigate OR target the sibblings

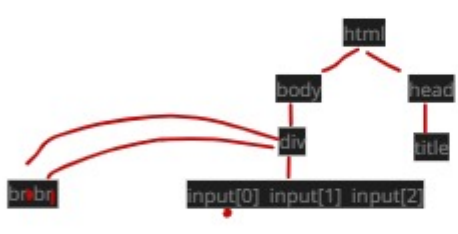
```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

xpath ==> sibblings

text()="I Phone 15"/following-sibbling

Xpath ==> Xpath is a path that is written by looking at the HTML tree structure

HTML code ==> Tree ==> HTML tree



===> Xpath

Xpath to enter the firstname
html/body/div/input[0] ==> FN

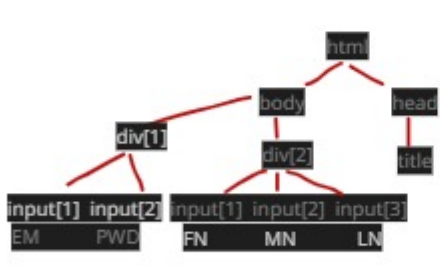
Xpath to enter the middlename
html/body/div/input[1] ==> MN

Absolute Xpath ==> Starts from the root element

```
await page.locator("xpath=/html/body/div/input[1]").fill("Raju")
await page.locator("xpath=/html/body/div/input[2]").fill("Ramu")
await page.locator("xpath=/html/body/div/input[3]").fill("Mary")
```

Relative Xpath

```
await page.locator("xpath=//input[1]").fill("Raju")
await page.locator("xpath=//input[2]").fill("Ramu")
await page.locator("xpath=//input[3]").fill("Mary")
```



locator(/html/body/div[1]/input[1]).fill("...")
locator(//div[1]/input[1]).fill("...")

1) Absolute Xpath
2) Relative Xpath
3) Xpath with tagname and Attribute

tagname[attr_name = "attr_value"]

//input[@id='email'] ==> XPATH

4) XPath with multiple attributes

5) Xpath with text()

//tagname[text() = "....."]

6) <button>Login In</button>
Xpath with contains()
//tagname[contains(text(), "....")]
//button[contains(text(), "Login"")] ==> Login In,
Login Later, ...

7)
<h1>Welcome</h1>

//*[attr_name = "attr_value"]
//*[text() = "....."]

```
<div class="product">
  <h2 class="title">Samsung S24</h2>
  <span class="price">₹80,000</span>
  <p class="desc">Flagship Samsung phone</p>
  <button class="buy">Buy Now</button>
</div>
```

```
await page.locator("//span[text()='₹80,000']/ancestor::div[@class='product']");
```

```
<div class="product">
  <h2 class="title">iPhone 15</h2>
  <span class="price">₹90,000</span>
  <p class="desc">Latest iPhone model</p>
  <button class="buy">Buy Now</button>
</div>
```

```
<div class="page">
  <section class="category">
    <div class="product-wrapper">
      <div class="product">
        <h2 class="title">Samsung S24</h2>
        <span class="price">₹80,000</span>
        <p class="desc">Flagship Samsung phone</p>
        <button class="buy">Buy Now</button>
      </div>
    </div>
  </section>

  <section class="category">
    <div class="product-wrapper">
      <div class="product">
        <h2 class="title">iPhone 15</h2>
        <span class="price">₹90,000</span>
        <p class="desc">Latest iPhone model</p>
        <button class="buy">Buy Now</button>
      </div>
    </div>
  </section>
</div>
```

```
await page.locator("//h2[text()='Samsung S24']/following::");
```

text = "Login"

CSS, XPATH and Playwright Specific

CSS Selectors

1) id locator
locator(email)

2) class locator
locator(email)

3) name locator
locator(tagname[name="....."])

Ex:
<input type="text" name="email" placeholder="email" />
<input type="password" name="pass" placeholder="pass" />
tagname => input
atrribute_name = attribute_value ===> attribute

4) attribute locator
locator([name="email"]) ==> email webelement
locator([type="password"]) ==> password webelement

5) tag name
locator(tagname) ==> locator(input)

6) multiple attributes

locator([attr_name = 'attr_value'][attr_name = 'attr_value'])

7) tagname with id
locator(tagnameid)
==> locator(input).nth(number) ==> email web element
==> locator(input#email) ==> email web element

8) tagname with class
locator(tagnameclass)

9) direct child selector(>)

10) descendant selector(space)
The above 2 selector will target child using the parent

```
<div class="card">
    <h2>Product Title</h2>
    <button>Buy</button>

    <div class="footer">
        <button>Add</button>
    </div>
</div>
```

Problems with CSS selectors(some places where CSS selectors wont work):-

1) We cannot use CSS to identify any web element using the text

```
<button>Buy</button>
<button>Add</button>
<h1>Welcome</h1>
```

locator("text='Welcome'")

playwright specific

2) When we want to navigate from child to parent in CSS

```
<div class="footer">
    Enter Email:<input type="email" >
</div>
```

locator(.footer > input).fill("....")

I want to target the parent starting from child

```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

Write the logic to click on the samsung s25 products buy button?

locator(button).nth(1) ==> Buy

```
<div class="product">
    <h2>Samsumg s25</h2>
    <button>Buy</button>
</div>
```

Solution is XPATH

1) text() = "Add" ==> Add button
<h1>Welcome</h1>
text() = "Welcome"

2)

```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

text() = 'Samsung s25'/../text() = 'Buy'

```
<div class="product">
    <h2>Samsumg s25</h2>
    <button>Buy</button>
</div>
```

3) We cannot use logical operators in CSS
and, or

<input type = "text" name = "email" placeholder = "email" />

locator(type="text" and name="email")
locator(type="text" or name="email")

==========
XPATH ==> locator(type="text" and name="email")
locator(type="text" or name="email")

4) We cannot use CSS selector to navigate OR target the sibblings

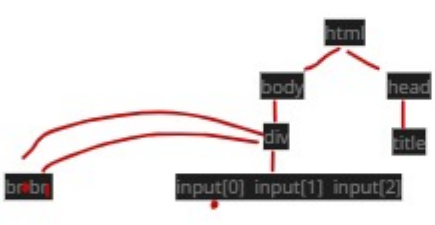
```
<div class="product">
    <h2>I Phone 15</h2>
    <button>Buy</button>
</div>
```

xpath ==> sibblings

text()="I Phone 15"/following-sibbling

Xpath ==> Xpath is a path that is written by looking at the HTML tree structure

HTML code ==> Tree ==> HTML tree



===> Xpath

Xpath to enter the firstname
html/body/div/input[0] ==> FN

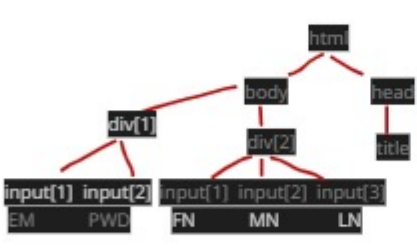
Xpath to enter the middlename
html/body/div/input[1] ==> MN

Absolute Xpath ==> Starts from the root element

```
await page.locator("xpath=/html/body/div/input[1]").fill("Raju")
await page.locator("xpath=/html/body/div/input[2]").fill("Ramu")
await page.locator("xpath=/html/body/div/input[3]").fill("Mary")
```

Relative Xpath

```
await page.locator("xpath=//input[1]").fill("Raju")
await page.locator("xpath=//input[2]").fill("Ramu")
await page.locator("xpath=//input[3]").fill("Mary")
```



locator(/html/body/div[1]/input[1]).fill("...")
locator(//div[1]/input[1]).fill("...")

1) Absolute Xpath
2) Relative Xpath
3) Xpath with tagname and Attribute

tagname[attr_name = "attr_value"]

//input[@id='email'] ==> XPATH

4) XPath with multiple attributes

5) Xpath with text()

//tagname[text() = '.....']

6) <button>Login In</button>
Xpath with contains()
//tagname[contains(text(), "....")]
//button[contains(text(), "Login")] ==> Login In,
Login Later, ...

7)

```
<h1>Welcome</h1>

//*[attr_name = "attr_value"]
//*[text() = "....."]
```

```
<div class="product">
  <h2 class="title">Samsung S24</h2>
  <span class="price">₹80,000</span>
  <p class="desc">Flagship Samsung phone</p>
  <button class="buy">Buy Now</button>
</div>

await page.locator("//span[text()='₹80,000']/ancestor::div[@class='product']");

<div class="product">
  <h2 class="title">iPhone 15</h2>
  <span class="price">₹90,000</span>
  <p class="desc">Latest iPhone model</p>
  <button class="buy">Buy Now</button>
</div>
```

```
<div class="page">
 <section class="category">
  <div class="product-wrapper">
   <div class="product">
    <h2 class="title">Samsung S24</h2>
    <span class="price">₹80,000</span>
    <p class="desc">Flagship Samsung phone</p>
    <button class="buy">Buy Now</button>
   </div>
  </div>
 </section>

 <section class="category">
  <div class="product-wrapper">
   <div class="product">
    <h2 class="title">iPhone 15</h2>
    <span class="price">₹90,000</span>
    <p class="desc">Latest iPhone model</p>
    <button class="buy">Buy Now</button>
   </div>
  </div>
 </section>
</div>
```

await page.locator("//h2[text()='Samsung S24']/following::*");

text = "Login"