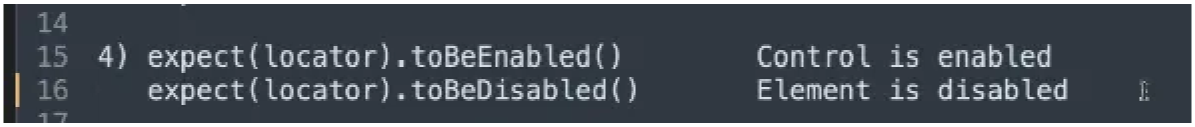
pavanpw-7

for checking enabled and disabled-



```
4) expect(locator).toBeEnabled()       Control is enabled
   expect(locator).toBeDisabled()      Element is disabled
```

to have attribute-

//6) **expect(locator).toHaveAttribute()** Element has attribute const regButton=await page.locator('#register-button') **await expect(regButton).toHaveAttribute('type','submit')**

pass in the attribute and the value of the attribute.

this is to match the full text-

await expect(await page.locator('.page-title h1')).toHaveText('Register') //full text

this is to match any partial text also-

await expect(await page.locator('.page-title h1')).toContainText('Reg') //partial text

to have value -

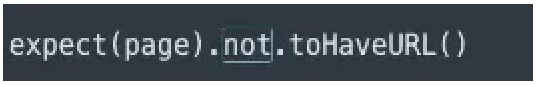
enter the value and then check if its entered and correct.

//9) **expect(locator).toHaveValue(value)** Input has a value const emailInput=await page.locator('#Email') await emailInput.fill('test@demo.com'); **await expect(emailInput).toHaveValue('test@demo.com')**

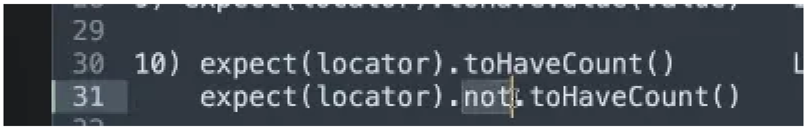
locator-

this word can be used for single element or multiple elements.

every assertion will have not also-

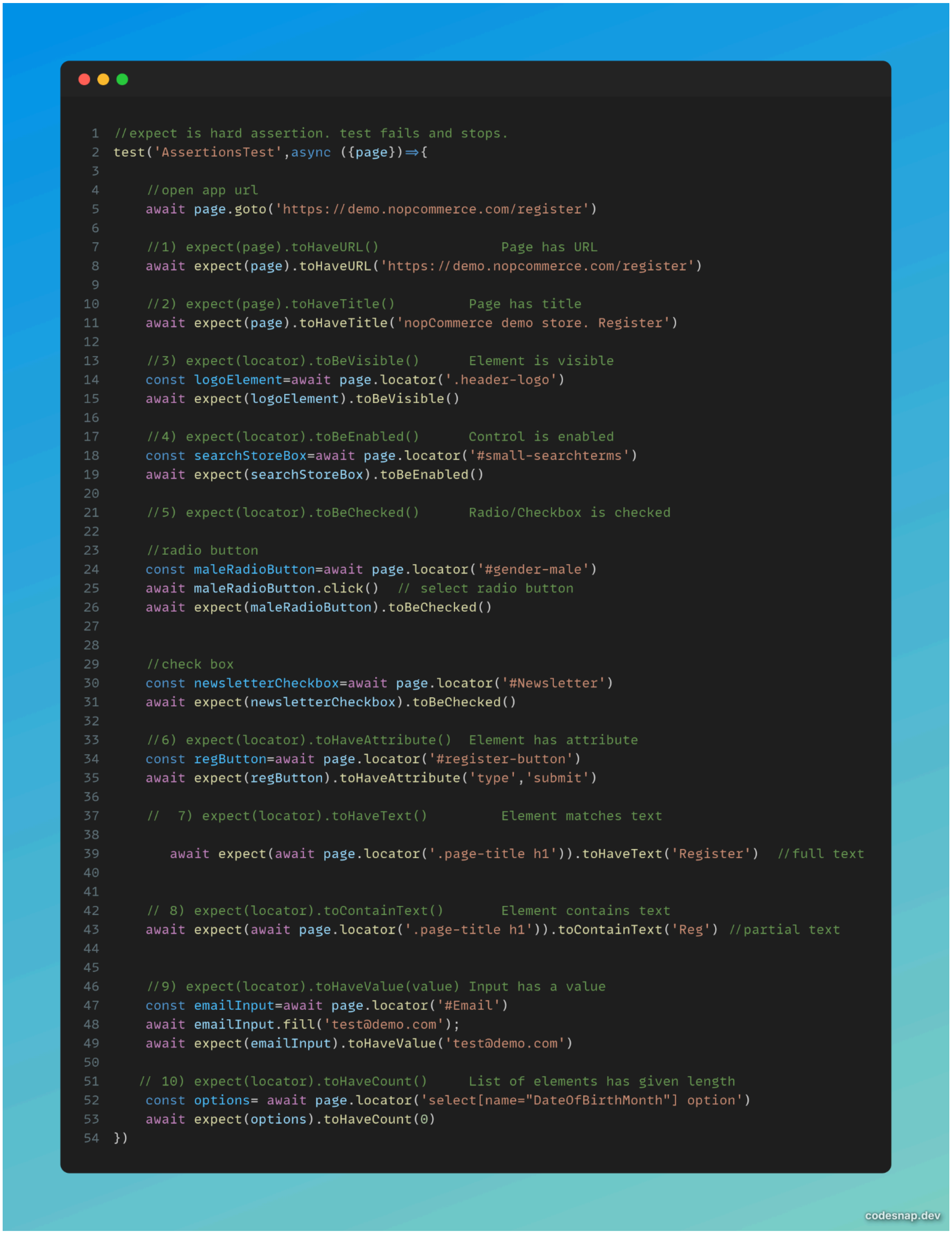


```
expect(page).not.toHaveURL()
```



```
10) expect(locator).toHaveCount()        L
    expect(locator).not.toHaveCount()
```

expect is hard assertion -



```
//expect is hard assertion. test fails and stops.
test('AssertionsTest',async ({page})=>{

    //open app url
    await page.goto('https://demo.nopcommerce.com/register')

    //1) expect(page).toHaveURL()               Page has URL
    await expect(page).toHaveURL('https://demo.nopcommerce.com/register')

    //2) expect(page).toHaveTitle()         Page has title
    await expect(page).toHaveTitle('nopCommerce demo store. Register')

    //3) expect(locator).toBeVisible()      Element is visible
    const logoElement=await page.locator('.header-logo')
    await expect(logoElement).toBeVisible()

    //4) expect(locator).toBeEnabled()      Control is enabled
    const searchStoreBox=await page.locator('#small-searchterms')
    await expect(searchStoreBox).toBeEnabled()

    //5) expect(locator).toBeChecked()      Radio/Checkbox is checked

    //radio button
    const maleRadioButton=await page.locator('#gender-male')
    await maleRadioButton.click()  // select radio button
    await expect(maleRadioButton).toBeChecked()


    //check box
    const newsletterCheckbox=await page.locator('#Newsletter')
    await expect(newsletterCheckbox).toBeChecked()

    //6) expect(locator).toHaveAttribute()  Element has attribute
    const regButton=await page.locator('#register-button')
    await expect(regButton).toHaveAttribute('type','submit')

    //  7) expect(locator).toHaveText()         Element matches text

       await expect(await page.locator('.page-title h1')).toHaveText('Register')  //full text


    // 8) expect(locator).toContainText()       Element contains text
    await expect(await page.locator('.page-title h1')).toContainText('Reg') //partial text


    //9) expect(locator).toHaveValue(value) Input has a value
    const emailInput=await page.locator('#Email')
    await emailInput.fill('test@demo.com');
    await expect(emailInput).toHaveValue('test@demo.com')

   // 10) expect(locator).toHaveCount()     List of elements has given length
    const options= await page.locator('select[name="DateOfBirthMonth"] option')
    await expect(options).toHaveCount(0)
})
```

codesnap.dev