pavanpw-19

webtable and pagination-

first capture the entire table -

const table=await page.locator('#productTable')

get total number of columns -

const columns= await table.locator('thead tr th')

get total number of rows-

const rows=await table.locator('tbody tr')

this is how we use filter to match a specific row-

//2) select check box for product 4 /* const machedRow= rows.filter({ has: page.locator('td'), hasText: 'Product 4' }) await machedRow.locator('input').check() */

filter has multiple options.

we are looking for page locators which have td tag.

and we are looking for product4.

storing in variable.

then use another locator for checkbox next to product4 and check it.

to capture a particular row-

const row=rows.nth(i);

example it will capture row one in first iteration then row two and so on.

to capture row wise data-

const tds=row.locator('td')

use this columns in the second for loop to read them.

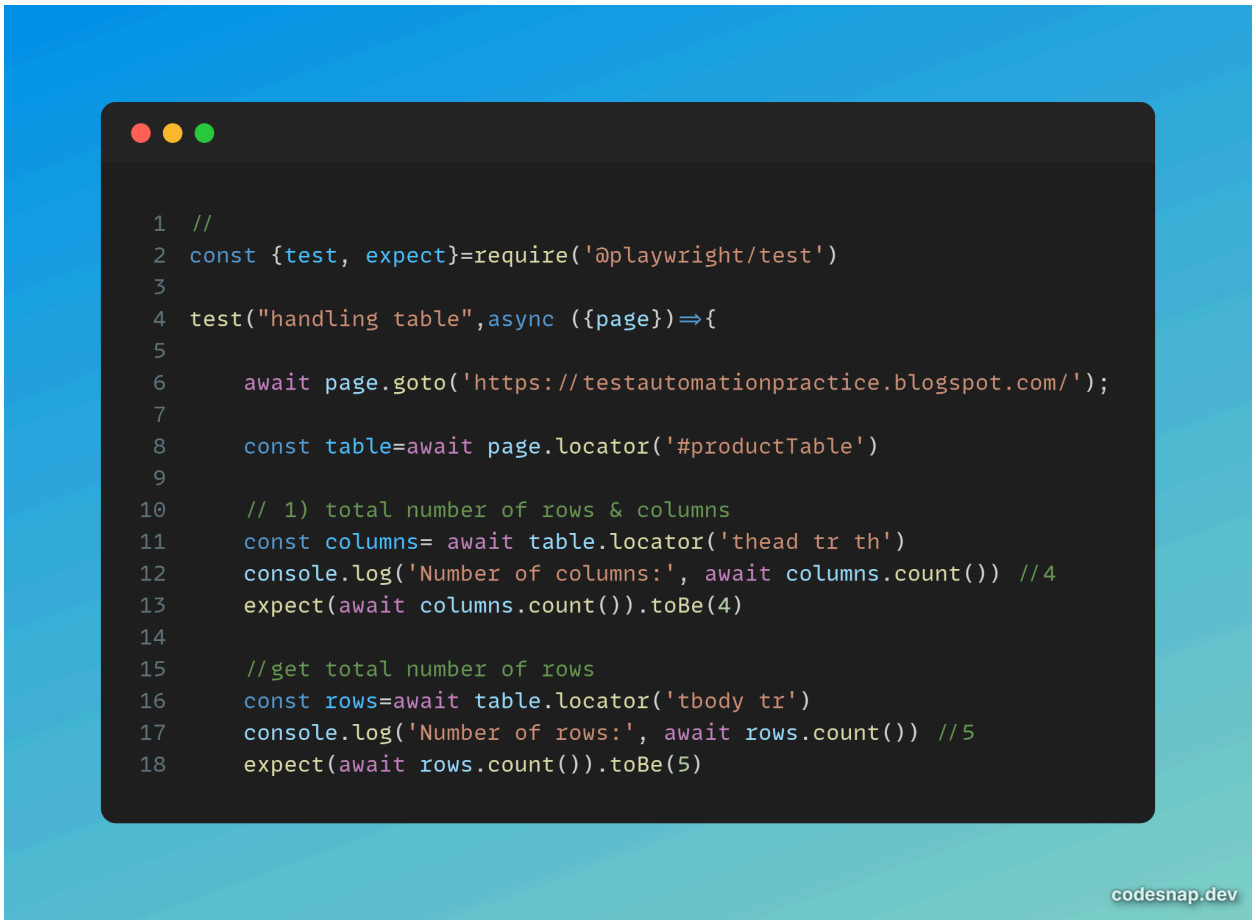
capture column wise data-

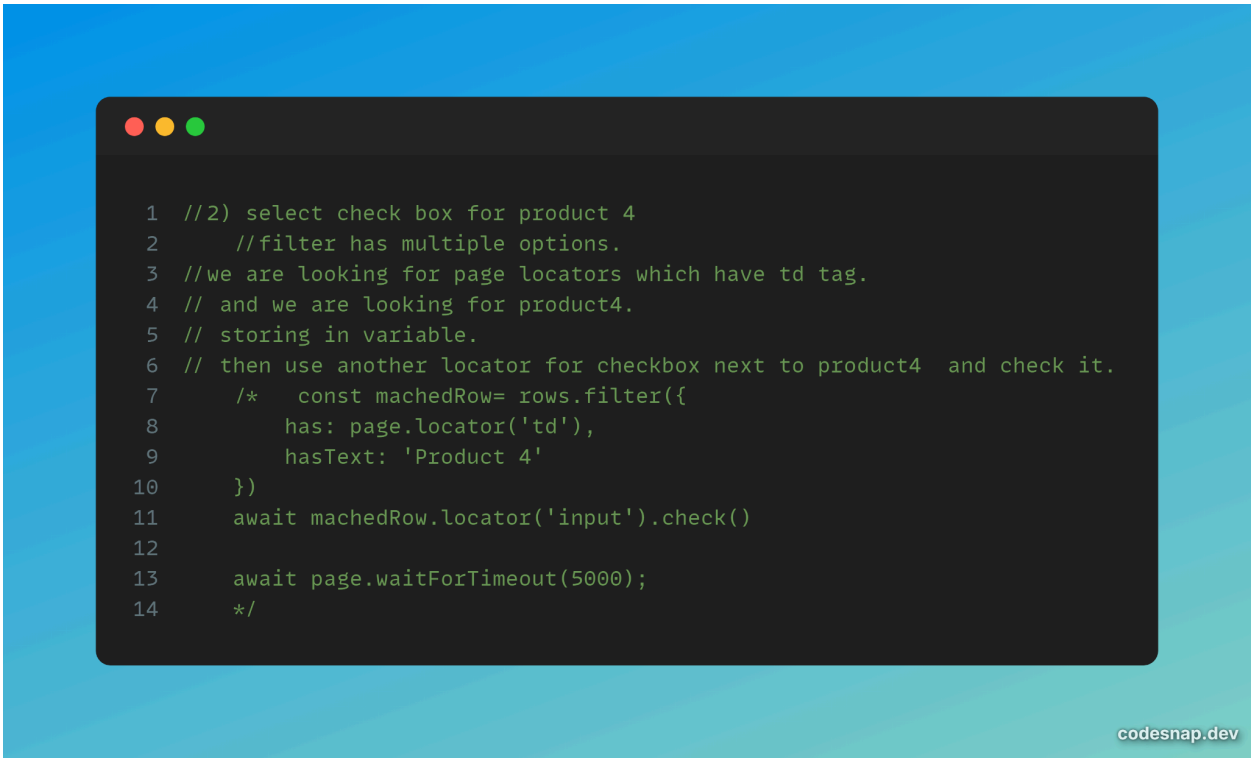
await tds.nth(j).textContent()

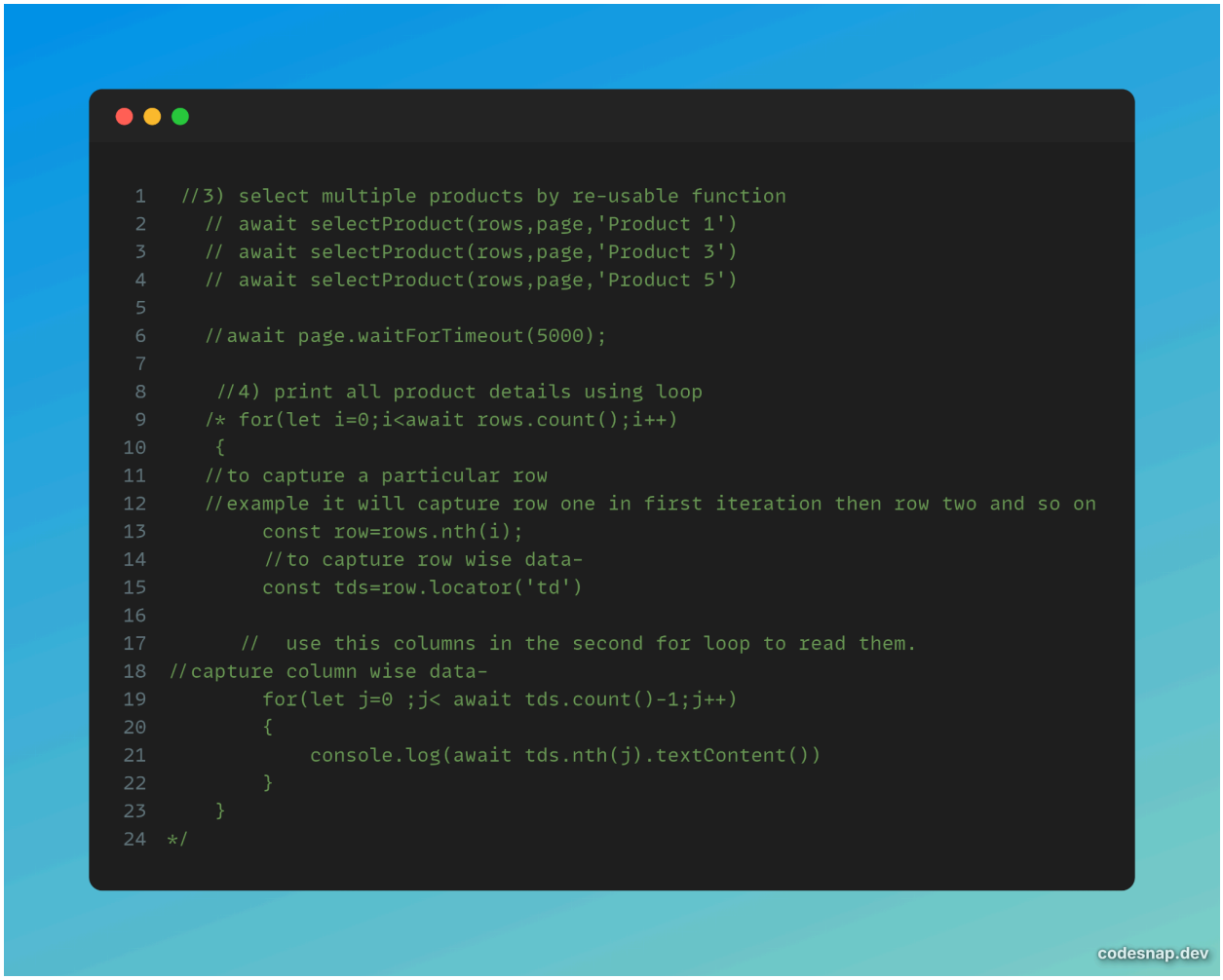
pagination-

first get number of pages.

const pages=await page.locator('.pagination li a')




//2) select check box for product 4
//filter has multiple options.
//we are looking for page locators which have td tag.
// and we are looking for product4.
// storing in variable.
// then use another locator for checkbox next to product4 and check it.
/* const machedRow= rows.filter({
has: page.locator('td'),
hasText: 'Product 4'
})
await machedRow.locator('input').check()
await page.waitForTimeout(5000);
*/
codesnap.dev



```
//3) select multiple products by re-usable function
  // await selectProduct(rows,page,'Product 1')
  // await selectProduct(rows,page,'Product 3')
  // await selectProduct(rows,page,'Product 5')

  //await page.waitForTimeout(5000);

   //4) print all product details using loop
  /* for(let i=0;i<await rows.count();i++)
   {
  //to capture a particular row
  //example it will capture row one in first iteration then row two and so on
       const row=rows.nth(i);
       //to capture row wise data-
       const tds=row.locator('td')

     //  use this columns in the second for loop to read them.
//capture column wise data-
       for(let j=0 ;j< await tds.count()-1;j++)
       {
           console.log(await tds.nth(j).textContent())
       }
   }
*/
```

```
//5) read data from all the pages in the table

//get total number of pages
    const pages=await page.locator('.pagination li a')
    console.log('Number of pages in the table:', await pages.count())

    for(let p=0 ;p< await pages.count(); p++)
    {
        if(p>0)
        {
            await pages.nth(p).click()
        }
        for(let i=0;i<await rows.count();i++)
        {
            const row=rows.nth(i);
            const tds=row.locator('td')

            for(let j=0 ;j< await tds.count()-1;j++)
            {
                console.log(await tds.nth(j).textContent())
            }
        }
        await page.waitForTimeout(3000);

    }

    await page.waitForTimeout(3000)
})


async function selectProduct(rows, page, name)
{
    const machedRow= rows.filter({
        has: page.locator('td'),
        hasText: name
    })
    await machedRow.locator('input').check()
}
```