Page Scrolling

```
import {chromium,test} from '@playwright/test';


test('webtable', async () => {
    const browser = await chromium.launch({ headless: false });
    const page = await browser.newPage();
    await page.goto('https://healthcaresuccess.com/case-studies');


      class PageScroller {
       constructor()
       {}


        async scrollBy(pixels:any) {
            await page.evaluate((scrollAmount) => {
                window.scrollBy(0, scrollAmount);
            }, pixels);
        }


        async scrollToBottom() {
            await page.evaluate(() => {
                window.scrollTo(0, document.body.scrollHeight);
            });
        }


 async scrollToElement(selector: string) {
  const locator = page.locator(selector).first(); // or .nth(0), .nth(1), etc.
  await locator.scrollIntoViewIfNeeded();
  await page.waitForTimeout(1000);
    }


    }
      const scr = new PageScroller();
      await page.pause();
    await scr.scrollBy(500);
    await scr.scrollToBottom();
    await scr.scrollToElement('a:has-text("About Us")');
   await page.waitForTimeout(2000);

});
```

DatePicker

```
import {test,chromium,expect} from '@playwright/test'
test('checkbox validation',async () => {

    const browser = await chromium.launch({headless:false});
    const context = await browser.newContext();
    const page = await context.newPage();
    await page.goto('https://jqueryui.com/resources/demos/datepicker/default.html');
    try{
    const day ='15';
    const month='May';
    const year ='2026';
    await page.pause();

    // write date picker logic
    await page.locator('#datepicker').click();
    await page.waitForSelector('.ui-datepicker-calendar');

    //year selection

    while(true)
    {
      const currentyear= await page.locator('.ui-datepicker-year').textContent();

      if(currentyear==year)
        break;
        await page.locator('.ui-datepicker-next.ui-corner-all').click();

    }

     //month selection

    while(true)
    {
      const currentmonth= await page.locator('.ui-datepicker-month').textContent();

      if(currentmonth==month)
```

```
      break;
      await page.locator('.ui-datepicker-next.ui-corner-all').click();


   }


   //Day selection
    await page.click(`.ui-datepicker-calendar a:has-text("${day}")`);
   const selectedDate = await page.$eval('#datepicker', input => (input as HTMLInputElement).value);
    console.log(`Selected date: ${selectedDate}`);
   }
   catch(error)
   {
     console.log(error)
   }


});
```

Image comparison:

```
import {test,chromium,expect} from '@playwright/test'
test('checkbox validation',async () => {


   const browser = await chromium.launch({headless:false});
   const context = await browser.newContext();
   const page = await context.newPage();
```

```
    await page.goto('https://the-internet.herokuapp.com/windows');


    await page.screenshot({path:'image1.png'});
    await page.waitForTimeout(2000);
    expect(await
page.screenshot()).toMatchSnapshot('image1.png',{maxDiffPixelRatio:0.2,});
 //expect(await page.screenshot()).toMatchSnapshot('healthcare-home.png', {
   // threshold: 0.2, // Adjust threshold for pixel difference tolerance
  //});


});
```

Fetching data from an API.

Reading or writing files

Database Queries

Delaying execution

window handles

```
import { test, chromium } from '@playwright/test';


test('Window handling example', async () => {
            const browser = await chromium.launch({ headless: false });
            const context = await browser.newContext();
            const page = await context.newPage();
```

```
      // Navigate to the test page
      await page.goto('https://the-internet.herokuapp.com/windows');

      // Listen for new page (window) event
      const [newPage] = await Promise.all([
    context.waitForEvent('page'),
    page.click('a[href="/windows/new"]') // Click the link that opens a new window
      ]);

      // Wait for the new page to load
      await newPage.waitForLoadState();

      // Log the title of the new window
  console.log('New window title:', await newPage.title());

      // Optionally, validate content in the new window
      const headerText = await newPage.locator('h3').textContent();
  console.log('Header in new window:', headerText);

      await browser.close();
});
```

Alerts:

```
import {test,chromium} from '@playwright/test'
test('JS Alerts', async () => {
```

```
const browser = await chromium.launch({headless:false});
 const context = await browser.newContext();
 const page = await context.newPage();



 await page.goto('https://the-internet.herokuapp.com/javascript_alerts');

  //await page.pause();
 page.once('dialog',async dialog => {

    console.log(`dialog title ::`,dialog.message())
    await dialog.accept()

 });
 await page.click('text=Click for JS Alert');

 await page.waitForTimeout(2000);
  page.once('dialog',async dialog => {

    console.log(`confirm dialog title ::`,dialog.message())
    await dialog.accept()

 });

 await page.click('text=Click for JS Confirm');
 await page.waitForTimeout(2000);
  page.once('dialog',async dialog => {


    console.log(`promt dialog title ::`,dialog.message())
    await dialog.accept('training')


 });


 await page.click('text=Click for JS Prompt');
 await page.waitForTimeout(2000);


})

===================================================================
```