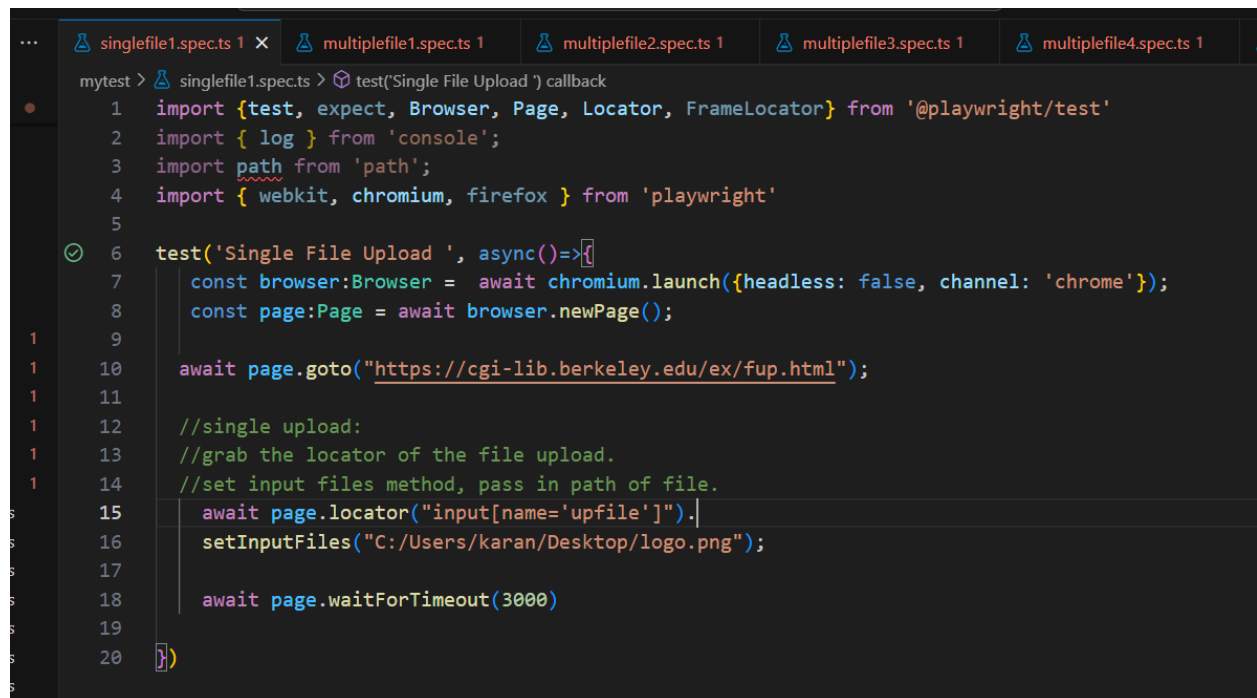
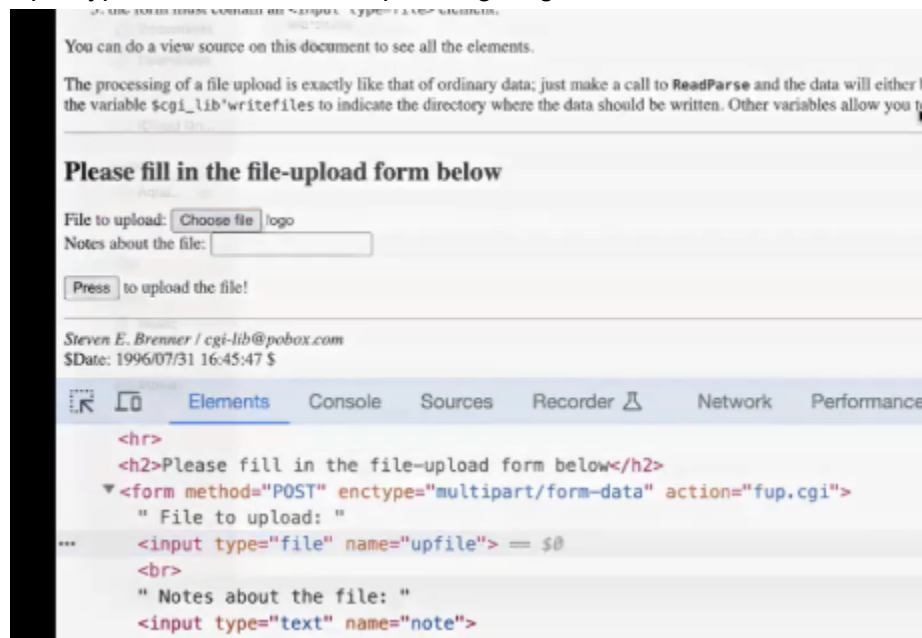


Input type should be file for uploading single file-



Multiple files-

Here also input type is file.

Pass multiple files-

```
... multipfile1.spec.ts 1 x multipfile2.spec.ts 1 multipfile3.spec.ts 1 multipfile4.spec.ts 1 multipfile5.spec.ts 1 mult

mytest > multipfile1.spec.ts > test('multiple File Upload ') callback
1 import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2 import { log } from 'console';
3 import path from 'path';
4 import { webkit, chromium, firefox } from 'playwright'
5
6 test('multiple File Upload ', async()=>{
7     const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8     const page:Page = await browser.newPage();
9
10    await page.goto("https://davidwalsh.name/demo/multiple-file-upload.php");
11
12    //multiple files:
13    //grab the locator of file upload.
14    //set input files method.
15    //pass in multiple files, give full path.
16    //use path.join method.
17    await page.locator("input[name='filesToUpload']")
18        .setInputFiles([
19        path.join("C:/Users/karan/Desktop/logo.png"),
20        path.join("D:/pramodpython/Links.txt"),
21        path.join("C:/Users/karan/Desktop/Karan_Atreyas_13years_Automation_Engineer.docx")]);
22
23    await page.waitForTimeout(4000);
24
25    })
```

Single file can also be sent this way-

```
... multipfile2.spec.ts 1 x multipfile3.spec.ts 1 multipfile4.spec.ts 1 multipfile5.spec.ts 1 multip

mytest > multipfile2.spec.ts > test('multiple File Upload ') callback
6 test('multiple File Upload ', async()=>{
7
8
9
10    await page.goto("https://davidwalsh.name/demo/multiple-file-upload.php");
11
12    //single file can also be sent this way.
13    //single file
14    //use path.join method.
15    //grab the locator of file upload.
16    //set input files method.
17    //pass in single file, give full path.
18    //use path.join method.
19    await page.locator("input[name='filesToUpload']")
20        .setInputFiles([
21        path.join("D:/pramodpython/Links.txt"),
22        ]);
23
24    await page.waitForTimeout(4000);
25
26    })
```

Deselect the files-

Multiple files deselected.

```
...  multiplefile4.spec.ts 1  multiplefile3.spec.ts 1 x  multiplefile5.spec.ts 1  multiplefile6.spec.ts 1
mytest >  multiplefile3.spec.ts >  test('multiple File Upload ') callback
1  import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2  import { log } from 'console';
3  import path from 'path';
4  import { webkit, chromium, firefox } from 'playwright'
5
6  test('multiple File Upload ', async()=>{
7      const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8      const page:Page = await browser.newPage();
9
10     await page.goto("https://davidwalsh.name/demo/multiple-file-upload.php");
11
12     await page.locator("input[name='filesToUpload']")
13         .setInputFiles([
14         path.join("C:/Users/karan/Desktop/logo.png"),
15         path.join("D:/pramodpython/Links.txt"),
16         path.join("C:/Users/karan/Desktop/Karan_Atreyas_13years_Automation_Engineer.docx")]);
17
18     await page.waitForTimeout(4000);
19
20     //deselect all the files.
21
22     //deselect files:
23     //grab the locator of the file upload.
24     //use set input files method.
25     //pass in empty array.
26     await page.locator("input[name='filesToUpload']").setInputFiles([]);
27
28     await page.waitForTimeout(4000);
29
30 }
```

Deselect single file –

```
multiplefile4.spec.ts 1 x multiplefile5.spec.ts 1 multiplefile6.spec.ts 1
mytest > multiplefile4.spec.ts > test('multiple File Upload ') callback
1 import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2 import { log } from 'console';
3 import path from 'path';
4 import { webkit, chromium, firefox } from 'playwright'
5
6 test('multiple File Upload ', async()=>{
7     const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8     const page:Page = await browser.newPage();
9
10    await page.goto("https://davidwalsh.name/demo/multiple-file-upload.php");
11
12    await page.locator("input[name='filesToUpload']")
13        .setInputFiles([
14        path.join("C:/Users/karan/Desktop/logo.png")
15    ])
16
17    await page.waitForTimeout(4000);
18
19
20
21    //deselect single files:
22    //grab the locator of the file upload.
23    //use set input files method.
24    //pass in empty array.
25    await page.locator("input[name='filesToUpload']").setInputFiles([]);
26
27    await page.waitForTimeout(4000);
28
29    })
```

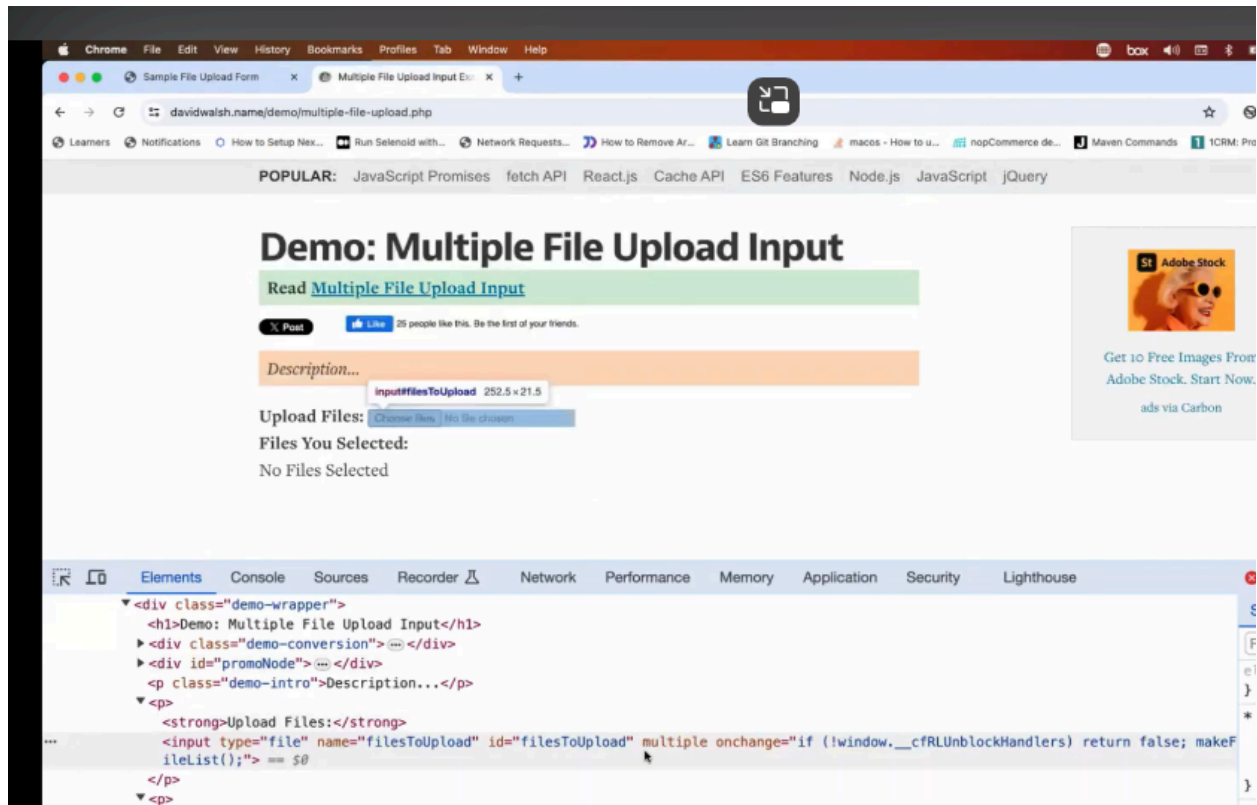
Create file on the fly after clicking on upload-  
Single file.

```
...  multiplefile5.spec.ts 1 x  multiplefile6.spec.ts 1
mytest >  multiplefile5.spec.ts > test('multiple File Upload ') callback
1  import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2  import { log } from 'console';
3  import path from 'path';
4  import { webkit, chromium, firefox } from 'playwright'
5
6  test('multiple File Upload ', async()=>{
7      const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8      const page:Page = await browser.newPage();
9
10     await page.goto("https://cgi-lib.berkeley.edu/ex/fup.html");
11
12     //create single file on fly.
13     //grab the locator of file chooser.
14     //set input files method.
15     //name of file with extention.
16     //mime type.
17     //buffer means what to write inside file.
18
19     await page.locator("input[name='upfile']").setInputFiles({
20         name: 'karan_resume.txt',
21         mimeType: 'text/plain',
22         buffer: Buffer.from('this is karan resume')
23     });
24
25     await page.waitForTimeout(14000);
26
27     })
```

Try with multiple files also.

```
multiplefile6.spec.ts 1 X
mytest > multiplefile6.spec.ts > test('multiple File Upload ') callback > name
1 import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2 import { log } from 'console';
3 import path from 'path';
4 import { webkit, chromium, firefox } from 'playwright'
5
6 test('multiple File Upload ', async()=>{
7     const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8     const page:Page = await browser.newPage();
9
10    await page.goto("https://davidwalsh.name/demo/multiple-file-upload.php");
11
12    //create multiple files on fly.
13    //grab the locator of file chooser.
14    //set input files method.
15    //name of file with extention.
16    //mime type.
17    //buffer means what to write inside file.
18
19    await page.locator("input[name='filesToUpload']").setInputFiles([
20        {
21            name: 'karan_resume.txt',
22            mimeType: 'text/plain',
23            buffer: Buffer.from('this is karan resume')
24        },
25        {
26            name: 'karan_resume.docx',
27            mimeType: 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
28            buffer: Buffer.from('this is karan resume in doc format')
29        }
30    ]);
31
32    await page.waitForTimeout(14000);
33
34    })
```

Attribute responsible for multiple file upload-



Input type should be file for file uploading -  
This is for single and multiple files.

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Sample File Upload FormMultiple File Upload Input Ex

cgilib.berkeley.edu/ex/fup.html

LearnersNotificationsHow to Setup Nex...Run Selenium with...Network Requests...How to Remove Ar...Learn Git Branch

**NOTE: This script is *not* operative on this server. You must install it on your own server in order to see it operate.**

This page allows you to upload a file. Note that the script will limit the size of the file to around 50k (so that the server doesn't get swamped with uploads).

The file upload form looks just like any other form except that:

1. the form tag must specify the POST method
2. the form tag must specify an enctype of multipart/form-data
3. the form must contain an <input type=file> element.

You can do a view source on this document to see all the elements.

The processing of a file upload is exactly like that of ordinary data; just make a call to `ReadParse` and the data will either be put in `%in` or some other variable. The variable `$cgi_lib'writefiles` to indicate the directory where the data should be written. Other variables allow you to further customize the script.

### Please fill in the file-upload form below

File to upload:

Choose file | AWS.txt

Notes about the file:

Press to upload the file!

Steven E. Brenner | cgilib@pobox.com  
\$Date: 1996/07/31 16:45:47 \$

ElementsConsoleSourcesRecorderNetworkPerformanceMemory

<p></p>

<p>The file upload form looks just like any other form except that: </p>

<ol></ol>

" You can do a view source on this document to see all the elements. "

<p></p>

<hr>

<h2>Please fill in the file-upload form below</h2>

<form method="POST" enctype="multipart/form-data" action="fup.cgi">

" File to upload: "

<input type="file" name="upfile">

<br>

" Notes about the file: "

<input type="text" name="note">

<br>

htmlbodyforminput