

[Home](#) > [puppeteer](#) > [Page](#) > [exposeFunction](#)

Page.exposeFunction() method

The method adds a function called `name` on the page's `window` object. When called, the function executes `puppeteerFunction` in node.js and returns a `Promise` which resolves to the return value of `puppeteerFunction`.

If the puppeteerFunction returns a `Promise`, it will be awaited.

NOTE: Functions installed via `page.exposeFunction` survive navigations.

Signature:

```
exposeFunction(name: string, puppeteerFunction: Function): Promise<void>;
```

Parameters

Parameter	Type	Description
name	string	Name of the function on the window object
puppeteerFunction	Function	Callback function which will be called in Puppeteer's context.

Returns:

Promise<void>

Example

An example of adding an `md5` function into the page:

```
const puppeteer = require('puppeteer');
const crypto = require('crypto');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  page.on('console', (msg) => console.log(msg.text()));
  await page.exposeFunction('md5', (text) =>
    crypto.createHash('md5').update(text).digest('hex')
  );
  await page.evaluate(async () => {
    // use window.md5 to compute hashes
    const myString = 'PUPPETEER';
    const myHash = await window.md5(myString);
    console.log(`md5 of ${myString} is ${myHash}`);
  });
  await browser.close();
})();
```

An example of adding a `window.readFile` function into the page:

```
const puppeteer = require('puppeteer');
const fs = require('fs');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  page.on('console', (msg) => console.log(msg.text()));
  await page.exposeFunction('readfile', async (filePath) => {
    return new Promise((resolve, reject) => {
      fs.readFile(filePath, 'utf8', (err, text) => {
        if (err) reject(err);
        else resolve(text);
      });
    });
  });
  await page.evaluate(async () => {
    // use window.readfile to read contents of a file
    const content = await window.readfile('/etc/hosts');
    console.log(content);
  });
  await browser.close();
})();
```