

Quick start

Getting Started

Installation

To use Puppeteer in your project, run:

```
npm i puppeteer  
# or "yarn add puppeteer"
```



Note: When you install Puppeteer, it downloads a recent version of Chromium (~170Mb Mac, ~282Mb Linux, ~280Mb Win) that is guaranteed to work with the API. To skip the download, see [Environment variables](#).

Usage

Note: Puppeteer requires at least Node v6.4.0, but the examples below use async/await which is only supported in Node v7.6.0 or greater.

Puppeteer will be familiar to people using other browser testing frameworks. You create an instance of **Browser**, open pages, and then manipulate them with Puppeteer's API.

Example - navigating to <https://example.com> and saving a screenshot as *example.png*:

Save file as **example.js**

```
const puppeteer = require('puppeteer');  
  
(async () => {  
  const browser = await puppeteer.launch();  
  const page = await browser.newPage();  
  await page.goto('https://example.com');  
  await page.screenshot({path: 'example.png'});  
  
  await browser.close();  
})();
```



Execute script on the command line

```
node example.js
```



Puppeteer sets an initial page size to 800px x 600px, which defines the screenshot size. The page size can be customized with [Page.setViewport\(\)](#).

Example - create a PDF.

Save file as **hn.js**

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://news.ycombinator.com', {waitUntil: 'networkidle2'});
  await page.pdf({path: 'hn.pdf', format: 'A4'});

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



Execute script on the command line

```
node hn.js
```



See [Page.pdf\(\)](#) for more information about creating pdfs.

Example - evaluate script in the context of the page

Save file as **get-dimensions.js**

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

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');

  // Get the "viewport" of the page, as reported by the page.
  const dimensions = await page.evaluate(() => {
    return {
      width: document.documentElement.clientWidth,
      height: document.documentElement.clientHeight,
      deviceScaleFactor: window.devicePixelRatio
    };
  });
})();
```



```
console.log('Dimensions:', dimensions);

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

Execute script on the command line

```
node get-dimensions.js
```



See [Page.evaluate\(\)](#) for more information on `evaluate` and related methods like `evaluateOnNewDocument` and `exposeFunction`.

Default runtime settings

1. Uses Headless mode

Puppeteer launches Chromium in headless mode. To launch a full version of Chromium, set the 'headless' option when launching a browser:

```
const browser = await puppeteer.launch({headless: false}); // default is true
```



2. Runs a bundled version of Chromium

By default, Puppeteer downloads and uses a specific version of Chromium so its API is guaranteed to work out of the box. To use Puppeteer with a different version of Chrome or Chromium, pass in the executable's path when creating a **Browser** instance:

```
const browser = await puppeteer.launch({executablePath: '/path/to/Chrome'});
```



See [Puppeteer.launch\(\)](#) for more information.

See [this article](#) for a description of the differences between Chromium and Chrome. [This article](#) describes some differences for Linux users.

3. Creates a fresh user profile

Puppeteer creates its own Chromium user profile which it **cleans up on every run**.

Next steps

- Learn more about [headless Chrome](#)

- Look over the examples.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 16, 2018.