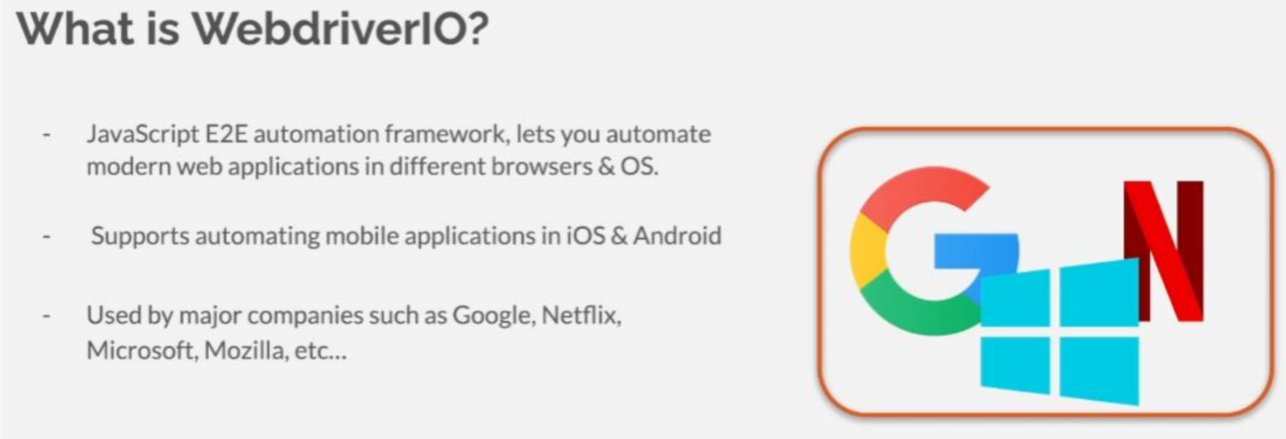
Contents

Section 1

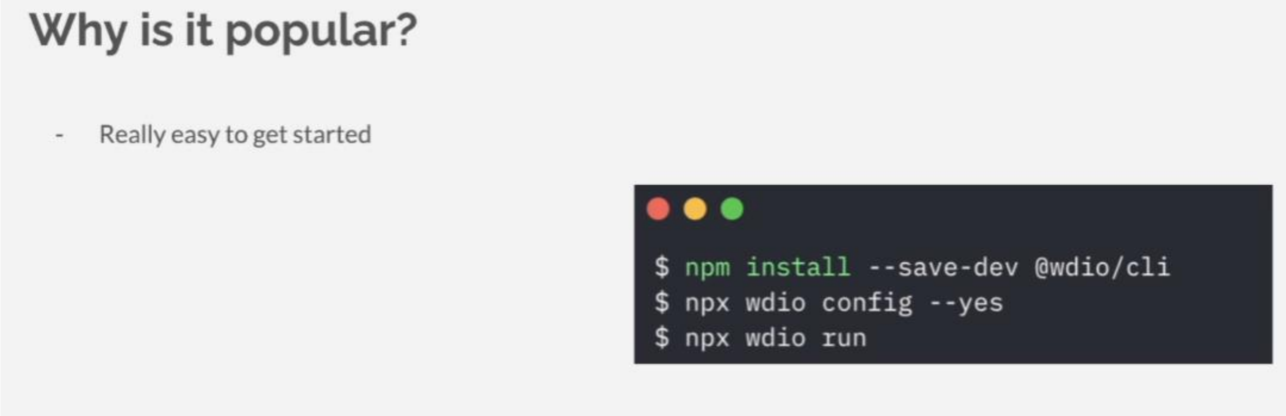
Lesson 3 What is WebdriverIO

- JS E2E automation framework



What is WebdriverIO?

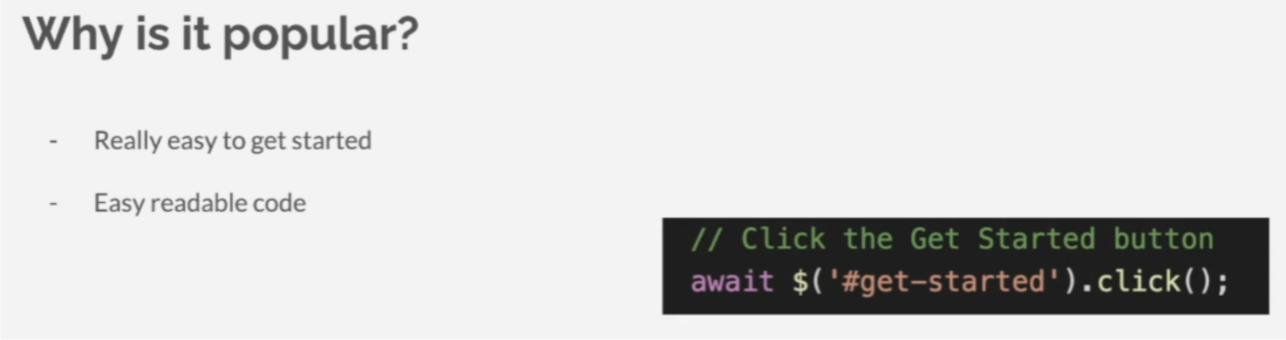
- JavaScript E2E automation framework, lets you automate modern web applications in different browsers & OS.
- Supports automating mobile applications in iOS & Android
- Used by major companies such as Google, Netflix, Microsoft, Mozilla, etc...



Why is it popular?

- Really easy to get started

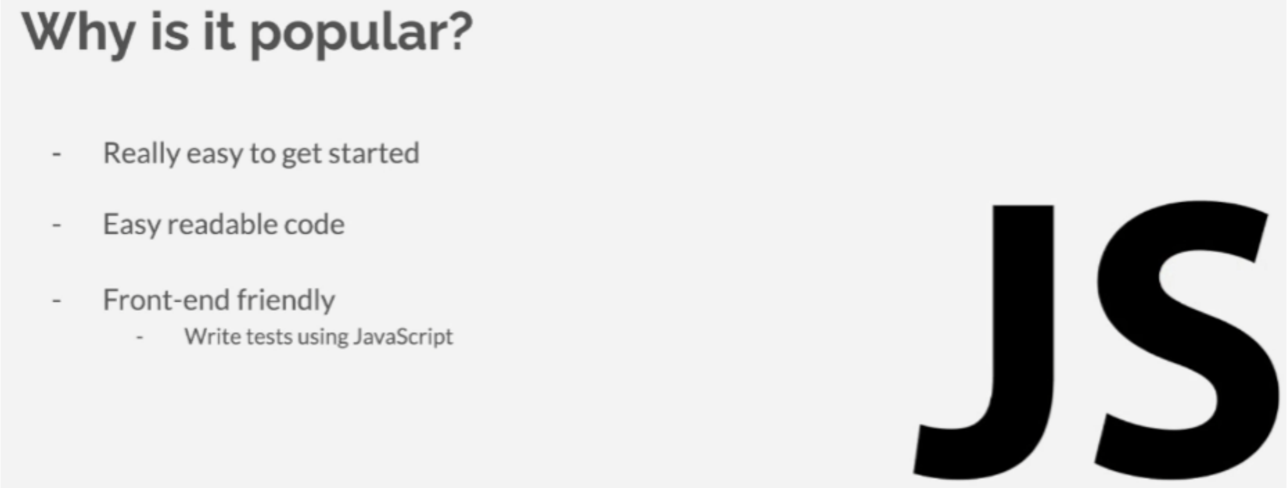
```
$ npm install --save-dev @wdio/cli
$ npx wdio config --yes
$ npx wdio run
```



Why is it popular?

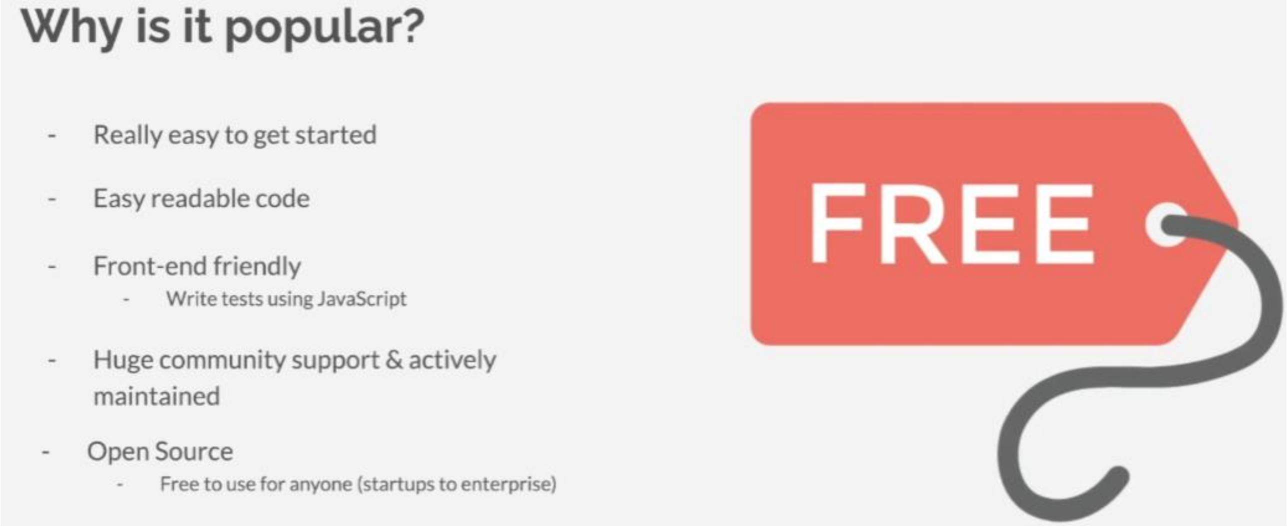
- Really easy to get started
- Easy readable code

```
// Click the Get Started button
await $('#get-started').click();
```



Why is it popular?

- Really easy to get started
- Easy readable code
- Front-end friendly
    - Write tests using JavaScript



Why is it popular?

- Really easy to get started
- Easy readable code
- Front-end friendly
    - Write tests using JavaScript
- Huge community support & actively maintained
- Open Source
    - Free to use for anyone (startups to enterprise)

Lesson 4 Getting Most out of the Course

Lesson 5 Discord Community

- https://discord.com/channels/1019379062013169734/@home

Lesson 8 Zipshare.com

- https://www.zipshare.com/

Section 2 Machine Setup

- Node
- VS Code installation
- Google Chrome
-

Section 3 WebdriverIO Setup & Installation

- codebase: https://github.com/dilpreetj/webdriverio-course

Lesson 13 WebdriverIO Project Setup

- npm init wdio ./path_to_new_project [npm init wdio ./wdio-course]
- this sigle command downloads the WebdriverIO CLI tool and runs a configuration wizard that helps to configure the test suite

Lesson 14 WebdriverIO Config Overview

Overview of wdio.conf.js

- baseURL,
- Specs,
- Cpabilities,
- Services,
- Framework,
- Reporter,
- Hooks

Section 4 Chrome 115+ Update Important

WebdriverIO v8.14.0+ Updates

- rm -rf node_modules package-lock.json
- npm i @wdio/cli@latest @wdio/local-runner@latest @wdio/mocha-framework@latest @wdio/spec-reporter@latest --save-dev
- in case updating older tests and conflicting with chrome browser drivers
- Chrome for testing: https://googlechromelabs.github.io/chrome-for-testing/

Section 5 Write Tests

Lesson 22 Write & Run first test

- Setup the folder structure
- Write a test for Home Page
    - Open URL & assert title (practice.automation.com)
- Run test: npx wdio

Lesson 23, 24 Fail test & Async

```
-   describe('Home', () => {
-       it('Open URL & assert title', async () => {
-           // Open URL
-           await browser.url('https://practice.sdetunicorns.com/');
-
-           // Assert title
-           await expect(browser).toHaveTitle('Practice E-Fail-Commerce Site –
    SDET Unicorns');
-           // await expect(browser).toHaveTitle('This is a random title');
-       })
-   })
```

Section 6 Working with Elements

Lesson 27 Finding Elements ($)

Lesson 28 & 29 Exercise using Xpath

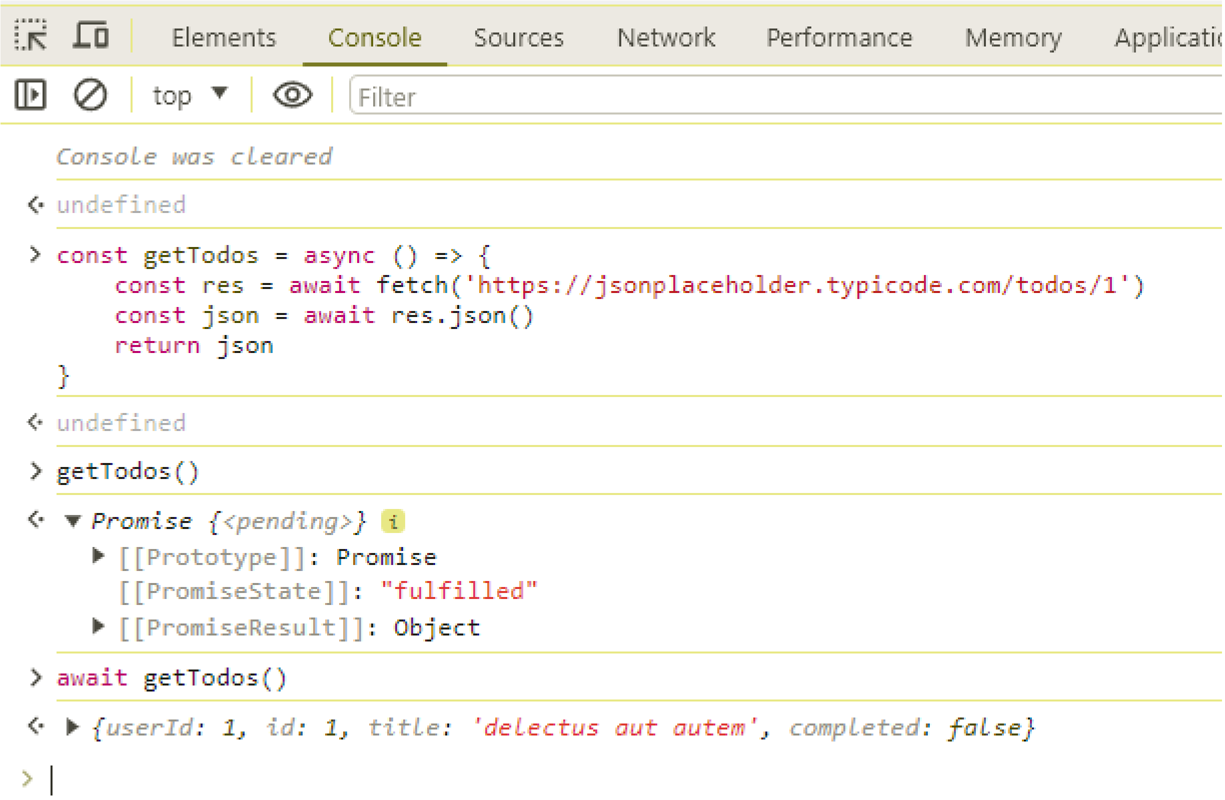
```
await $('//img[@alt="Practice E-Commerce Site"]').click();
```

Section 22 JavaScript Basics

const getTodos = async () => {
  const res = await fetch('https://jsonplaceholder.typicode.com/todos/1')
  const json = await res.json()
  return json
}

getTodos()

await getTodos()



```
Console was cleared
< undefined
> const getTodos = async () => {
      const res = await fetch('https://jsonplaceholder.typicode.com/todos/1')
      const json = await res.json()
      return json
  }
< undefined
> getTodos()
< ▼ Promise {<pending>} i
    ▶ [[Prototype]]: Promise
      [[PromiseState]]: "fulfilled"
    ▶ [[PromiseResult]]: Object
> await getTodos()
< ▶ {userId: 1, id: 1, title: 'delectus aut autem', completed: false}
> |
```

with [Shift + Enter] you can add a line to chrome console

Section 23 Quick HTML and CSS refresher

Lesson 97 HTML Overview

- https://replit.com/@automationbro/Introduction-to-HTML-CSS#index.html

Lesson 98 CSS Overview

Lesson 99 HTML DOM

Lesson 100 Custom CSS Selectors

- https://gist.github.com/magicznyleszek/809a69dd05e1d5f12d01

List of Selectors

- id (#id)
- Class (.class)
- Attribute (h1)
- Custom CSS (input[type="submit"])

Custom CSS Selector

- Attribute selectors
  - Exact, contains, begins with, etc..
- Contextual selectors
  - Descendant, child, sibling etc..
- Pseudo-class selectors
  - a:link, p:hover etc..
- Pseudo-class for siblings:
  - First-child, only-child, first-of-type etc..

p[class='paragraph'] a

- https://practice.sdetunicorns.com/
- #navigation>li>a:first-child
- #navigation>li>a:last-child
- [id="blog-link"]
- .main-menu .linkback a:last-child
- .main-menu .linkback a:nth-child(2)
- a[href="index.html"]
- a[href^="index"]
- a[href$="html"]
-

Lesson 101 How to use XPath

What is XPath

- XML path to navigate through HTML DOM
- Syntax
  - Xpath = //tagname[@Attribute='Value']
- Types of XPath
  - Absolute (never use)
  - /html/body//div[2]/div/[1]/div/h4[1]/b/html[1]/body[1]
- Relative
  - //tagname[@Attribute='Value']

List of Selectors

- Basic Xpath (//input[@type=’submit’]
- Contains (//*[contains(@href, ‘index’)])
- Or & And (//*[@href, ‘index.html’ and @role=’menuitem’])
- Starts with (//a[starts-with(@href,’index’)])
- Text (//a[text()=’About’])

Examples to HomePage:

//img[@src="https://cdn.pixabay.com/photo/2017/02/01/22/02/mountain-landscape-2031539_960_720.jpg"]

//*[contains(@href,'index')]

//*[@href='index.html' and @role='menu-item']

//a[starts-with(@href,'index')]

//a[text()='Backup']