FREE eBook

LEARNING google-chrome

Free unaffiliated eBook created from
Stack Overflow contributors.

#google-chrome

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: google-chrome

It is an unofficial and free google-chrome ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-chrome.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with google-chrome

Remarks

Google Chrome is a web browser available on desktop (Windows, ChromeOS, MacOS, Linux) and mobile (Android, iOS) platforms.

Web Developers should understand Chrome's frequent release cycle, multiple version channels, developer tools, and extension apis.

Examples

Installation or Setup

Install Chrome by first considering which **version** to use, relevant if you are going to be doing Web Development:

Google Chrome releases major versions about every 6 weeks and provides access to multiple versions of the product as they are being stabilized. This process is done in a series of channels.

- **Stable Channel** is the version released to all users, it is the default installation experience. It receives security updates as needed.
- **Beta Channel** is where the next major version is stabilized for approximately 1 month before moving to Stable. It is updated roughly weekly.
- **Dev Channel** is very close to the latest code, with minimal testing. It is updated 1 or 2 times a week.

A desktop installation of chrome can be moved from Stable to Beta or Dev in Chrome's preferences.

The **Canary** build undergoes minimal testing and is released daily. It provides access to the latest code and can be installed at the same time as a stable, beta, and dev versions of Chrome.

Install by following links to your choice of Stable (and then modifying it to Beta or Dev) or Canary. On Android and iOS, use the respective Play / App Store.

Steps to install:

1. Downloading Chrome For PC/Mac/Linux

2. Click "Download Chrome". This will open the Terms of Service window.

3. Determine if you want Chrome as your default browser.

- If you set it as the default browser, it will open whenever a link for a webpage is clicked in another program, such as email.
- You can opt to send usage data back to Google by checking the box labeled “Help make Google Chrome better…” This will send back crash reports, preferences and button clicks. ***It does not send any personal information or track websites.***

4. Click “Accept and Install” after reading the Terms of Service. The installer will start and you will have Google Chrome installed when it has finished.

Extensions

Google Chrome supports **extensions** that augment the way the browser works. They can add functionality to web pages or to the browser UI. Any developer can create an extension and list it in the Chrome Web Store.

See more on the extensions page.

Read Getting started with google-chrome online: https://riptutorial.com/google-chrome/topic/4203/getting-started-with-google-chrome

Chapter 2: Chrome DevTools

Remarks

Chrome DevTools product documentation.

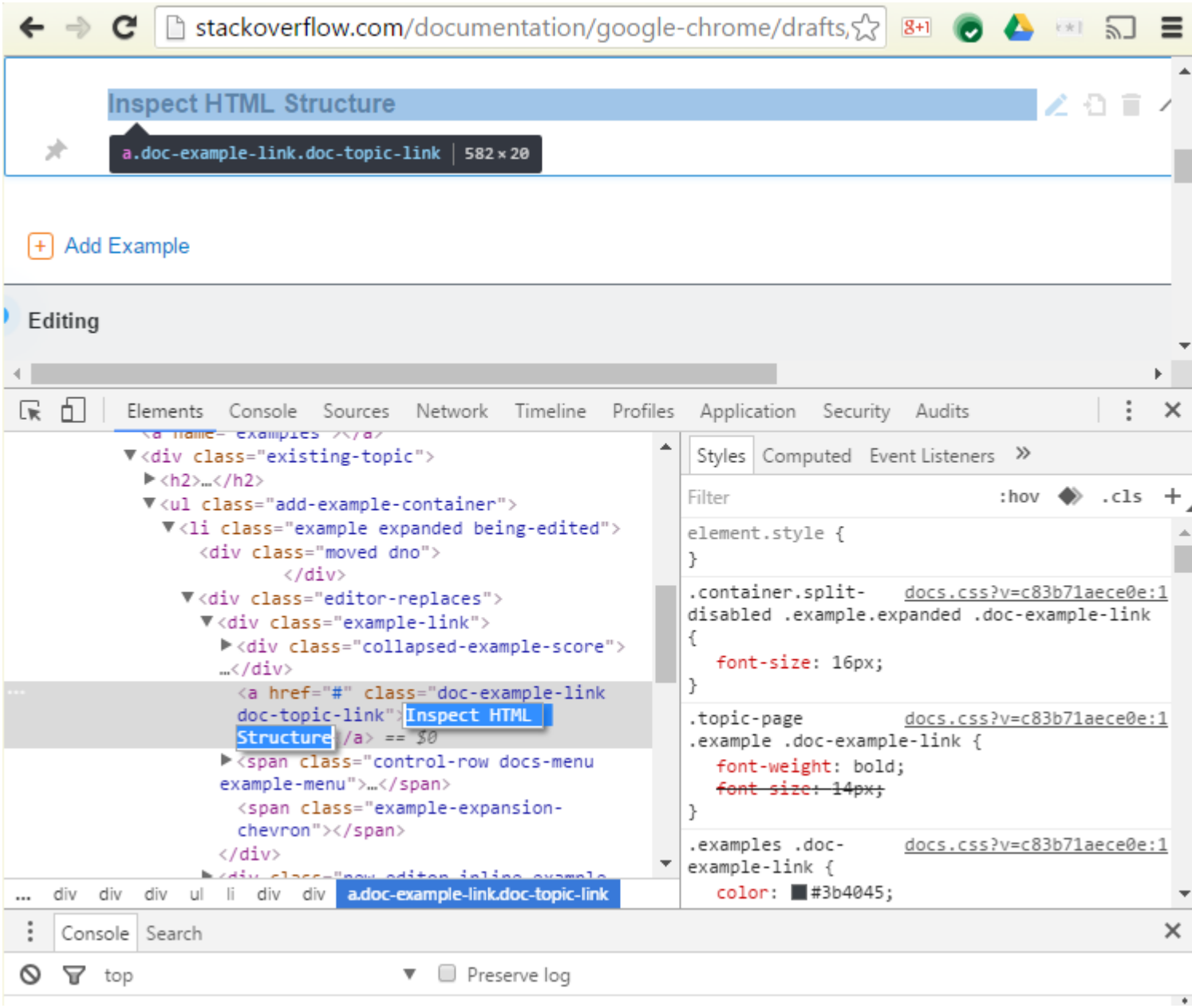
Examples

Inspect HTML Structure

On a desktop version of Chrome, the contents of the page can be inspected. This shows the document object model (DOM) of the HTML, the Cascading Style Sheet styles (CSS), and much more.

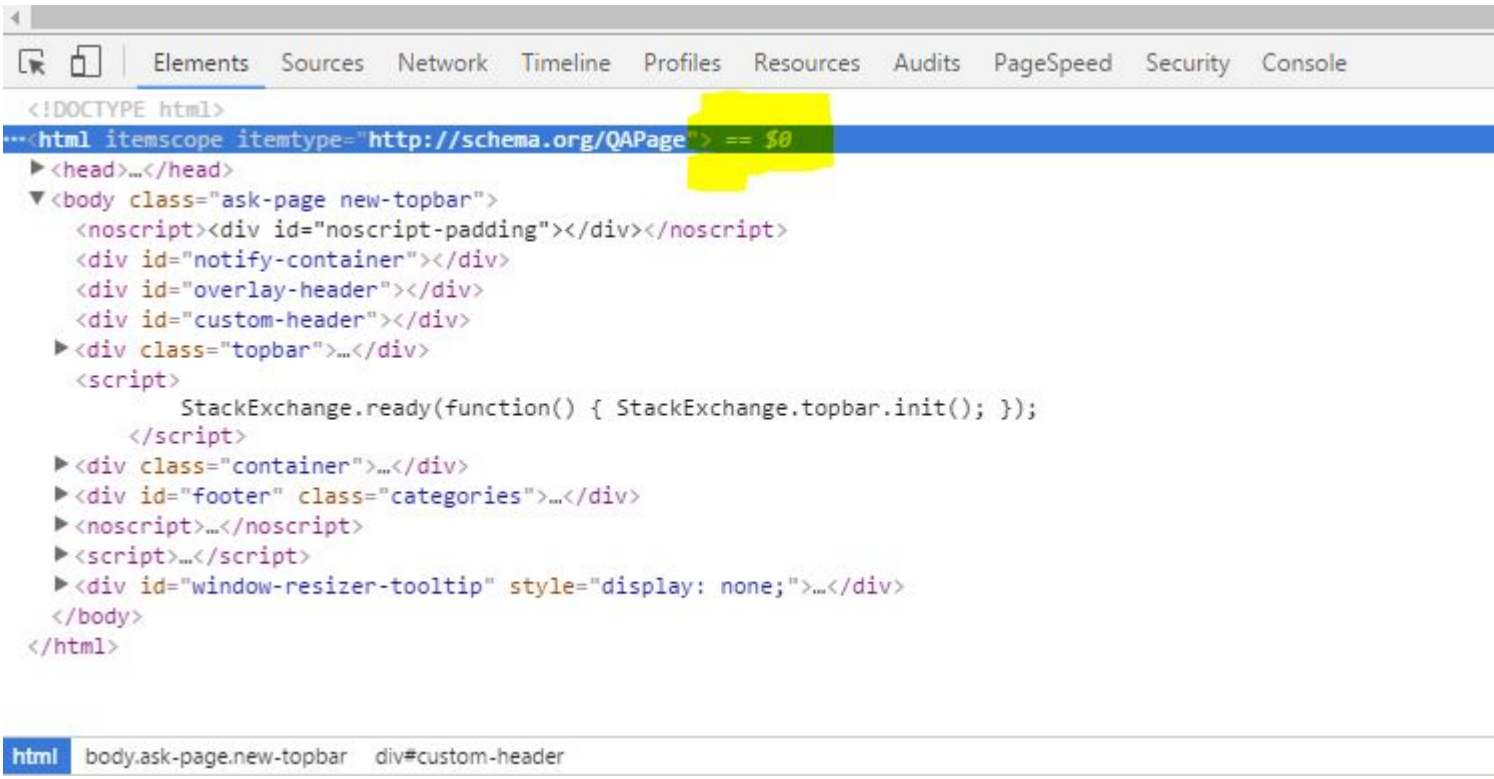
Enter inspection by one of many options:

- Right click on a web page, and select **Inspect**
- From the Chrome Menu, select **More Tools**, **Developer Tools**
- Use a keyboard shortcut, e.g. Ctrl+Shift+I on Windows.



DOM node index($0, $1, etc..)

In Google Chrome's developer tools "Elements", you can see that the selected line shows `==$0` that is the DOM node index(as shown below):



> $0 returns the most recently selected element or JavaScript object, $1 returns the second most recently selected one, and so on.

This is useful in debugging. The $0, $1, $2, $3 and $4 commands work as a historical reference to the last five DOM elements inspected within the Elements panel or the last five JavaScript heap objects selected in the Profiles panel.

Read Chrome DevTools online: https://riptutorial.com/google-chrome/topic/6762/chrome-devtools

Chapter 3: Chrome Extensions

Remarks

Google Chrome supports extensions that augment the way the browser works. They can add functionality to web pages or to the browser UI.

Examples

Browser Action running executeScript on a page.

manifest.json

```
{
  "name": "Hello Page",
  "description": "Add 'Hello' to the current page.",
  "version": "1.0",
  "permissions": [
    "activeTab"
  ],
  "background": {
    "scripts": ["background.js"],
    "persistent": false
  },
  "browser_action": {
    "default_title": "Say Hello on this page"
  },
  "manifest_version": 2
}
```

background.js

```
chrome.browserAction.onClicked.addListener(function(tab) {
  chrome.tabs.executeScript({
    code: 'document.body.insertAdjacentText("beforeBegin", "Hello!")'
  });
});
```

Read Chrome Extensions online: https://riptutorial.com/google-chrome/topic/4441/chrome-extensions

Chapter 4: Headless browsing

Introduction

Chrome supports headless browsing that's exposed through a switch that can be used when starting the process. This enables opening pages without creating a browser window, thus a graphical environment is not required.

When used in conjuction with an appliance that connects to the remote debugging port, it enables interacton with the document, which is particularly useful for testing and CI automation, where a graphical environment is not necessary to get results or isn't available.

Syntax

- chrome --headless $SWITCHES https://stackoverflow.com

Remarks

Historically, other's have succeeded at using Chrome as a headless browser by running the process in a hidden display.

Invoking Chrome directly is not the only available option for using it as a headless browser. The Embedder API also alows one to use Chrome directly from the application's process.

Examples

Taking screenshots

The following will produce a PNG image in the current directory of the loaded page.

```
chrome --headless --screenshot https://stackoverflow.com
```

Interacting with documents

Using the `--remote-debugging-port` to expose a debugger accessible over HTTP is one way appliances can connect and interact with the document using the Chrome Debugging Protocol.

```
chrome --headless --remote-debugging-port=9222 https://stackoverflow.com
```

You can then navigate to http://localhost:9222 and use Chrome DevTools interactively.

Read Headless browsing online: https://riptutorial.com/google-chrome/topic/8619/headless-browsing

Credits

S. No	Chapters	Contributors
1	Getting started with google-chrome	Ani Menon, Avi W., Community, Stephen Leppik, Vincent Scheib
2	Chrome DevTools	Ani Menon, Vincent Scheib
3	Chrome Extensions	Vincent Scheib
4	Headless browsing	Filip Dupanović