

=====

Tutor: Rocco Balsamo

Reference: UDEMY

Course: Devtools Pro: The Basics of Chrome Developer Tools

=====

1. Course URL:

<https://www.udemy.com/course/devtools-2017-the-basics-of-chrome-developer-tools/>

2. Document prepared by: **Rajat Verma**

- a. <https://www.linkedin.com/in/rajat-v-3b0685128/>
- b. <https://github.com/rajatt95>
- c. <https://rajatt95.github.io/>

3. Other Documents:

- a. <https://github.com/rajatt95/Documents>
-

Softwares:

1. Browser - Chrome
 2. Programming language - Javascript
-

1. Learnings from Course (UDEMY - RB - Devtools Pro: The Basics of Chrome Developer Tool)

a. Links:

- i. Application:
 1. <https://todomvc.com/examples/vanillajs/>
 2. <https://www.yelp.com/>
 3. <https://badssl.com/>
-

b. Tabs:

i. Elements

1. DOM

ii. Console

1. We can write Javascript commands

iii. Sources

1. The source code of the application goes here
2. This is helpful to check the application behavior/flow
 - a. How exactly our code is working
 - b. We can put the Breakpoints

- i. Perform a click on the line number
 - 3. **Snippets:**
 - a. This is helpful to save a snippet of code for usage in future
 - iv. **Audit (Old) -> Performance Insights**
 - 1. This gives a bunch of tips
 - a. How can we get our page to load a little bit quicker.
 - v. **Network**
 - 1. This panel shows all the Network Requests and Response details
 - a. Headers
 - b. Payload
 - c. Preview
 - d. Response
 - e. Timing
 - vi. **Performance**
 - vii. **Memory**
 - viii. **Application**
 - 1. This tab has the information regarding
 - a. Local Storage
 - b. Session storage
 - c. Cookies
 - ix. **Security**
 - 1. This tab gives information regarding the Security of the web page.
 - x. **Lighthouse**
 - xi. **Recorder**
 - xii. **AdBlock**
 - xiii. **More Tools**
-

- c. **Features:**
 - i. Steps to get the HTML located on the web page:
 - 1. Right click
 - 2. View Page Source
 - ii. Inspect Element
 - iii. Dock side
 - iv. Change the text of an element
 - v. Change the styling of an element
 - vi. Toggle Device Toolbar:
 - vii. Using the Console to Run Javascript
 - viii. Keyboard Shortcuts
 - ix. Styling Editors
-
-

=====Section 1: The Basics=====

-----1. What We'll Accomplish -----

1. The Basics

- a. Introduction & The Elements Panel
- b. A Quick Tour of DevTools
- c. Using the Console to Run Javascript
- d. Sources Tab and Basic Debugging
- e. Snippets: The Console on Steroids

2. Digging Deeper

- a. Cookies and Local Storage with the Application Tab
 - b. A Peak at Network Calls
 - c. Page Load Speed with Audits
 - d. Performance, Memory, and Security
 - e. The Most Useful Keyboard Shortcuts
 - f. Styling Editors
-

3. Detailed Lecture Descriptions

a. Introduction & The Elements Panel:

- i. Tweak HTML tags and styling with the elements panel. A Quick Tour of DevTools: Run through each of the tabs in DevTools and understand the basics about what each of them does.

b. Using the Console to Run Javascript:

- i. Run simple Javascript commands in the console panel. Sources Tab and Basic Debugging: Learn the basics of adding breakpoints to stop and step through your code. Also known as debugging!

c. Snippets:

- i. The Console on Steroids: Learn how to test out larger chunks of code in the snippets panel.

d. Cookies and Local Storage with the Application Tab:

- i. Inspect and modify cookies and local storage using the application tab.

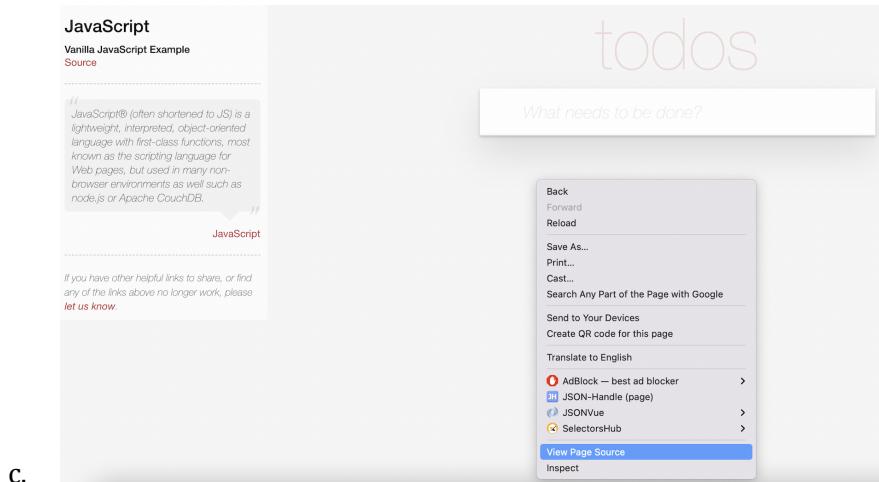
e. A Peak at Network Calls:

- i. A look at the network calls from the yelp reservation page.
-

-
- f. Page Load Speed with Audits:**
 - i. Understand how to review page load issues with the Audits panel. And a note about the IRS.
 - g. Performance, Memory, and Security:**
 - i. A quick dive into the more advanced panels in dev tools: Performance / Timeline, Memory, and Security.
 - h. The Most Useful Keyboard Shortcuts:**
 - i. A look at the keyboard shortcuts that I find most helpful in navigating around in Chrome Devtools.
 - i. Styling Editors:**
 - i. A lecture from my follow-up course that takes a deeper dive into how to use the different styling tools available in DevTools.

-----2. Introduction & The Elements Panel -----

1. Application:
 - a. <https://todomvc.com/examples/vanillajs/>
2. Steps to get the HTML located on the web page:
 - a. Right click
 - b. View Page Source



```

1  <!doctype html>
2  <html lang="en" data-framework="javascript">
3      <head>
4          <meta charset="utf-8">
5          <title>VanillaJS • TodoMVC</title>
6          <link rel="stylesheet" href="node\_modules/todomvc-common/base.css">
7          <link rel="stylesheet" href="node\_modules/todomvc-app-css/index.css">
8      </head>
9      <body>
10         <section class="todoapp">
11             <header class="header">
12                 <h1>todos</h1>
13                 <input class="new-todo" placeholder="What needs to be done?" autofocus>
14             </header>
15             <section class="main">
16                 <input id="toggle-all" class="toggle-all" type="checkbox">
17                 <label for="toggle-all">Mark all as complete</label>
18                 <ul class="todo-list"></ul>
19             </section>
20             <footer class="footer">
21                 <span class="todo-count"></span>
22                 <ul class="filters">
23                     <li>
24                         <a href="#" class="selected">All</a>
25                     </li>
26                     <li>
27                         <a href="#/active">Active</a>
28                     </li>
29                     <li>
30                         <a href="#/completed">Completed</a>
31                     </li>
32                 </ul>
33                 <button class="clear-completed">Clear completed</button>
34             </footer>
35         </section>
36         <footer class="info">
37             <p>Double-click to edit a todo</p>
38             <p>Created by <a href="http://twitter.com/oscargodson">Oscar Godson</a></p>
39             <p>Refactored by <a href="https://github.com/cburgmer">Christoph Burgmer</a></p>
40             <p>Part of <a href="http://todomvc.com">TodoMVC</a></p>
41         </footer>

```

d.

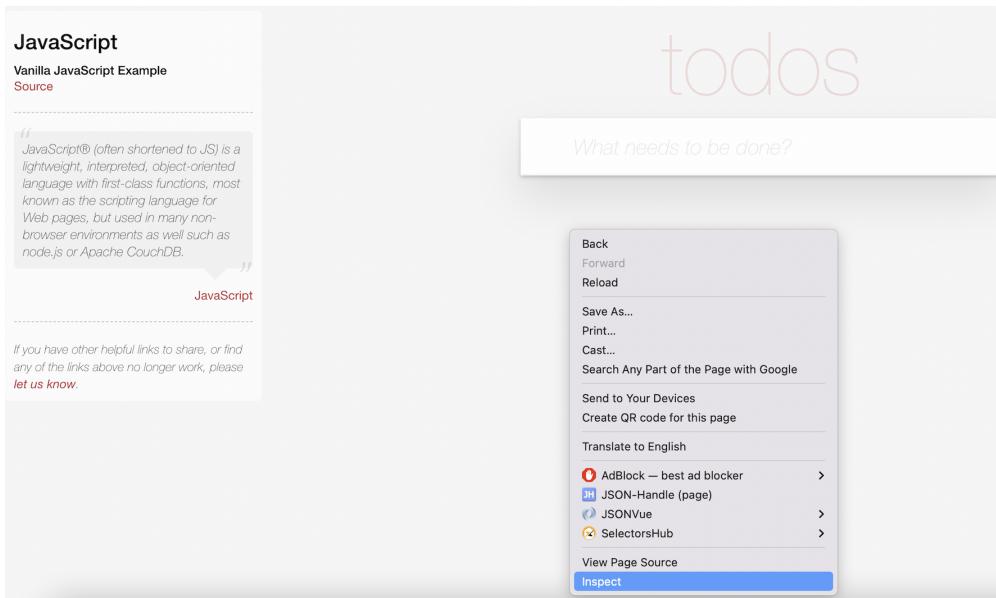
3. But it does not give us everything

- a. That we need to change and modify live web pages
 - b. To run console commands
-

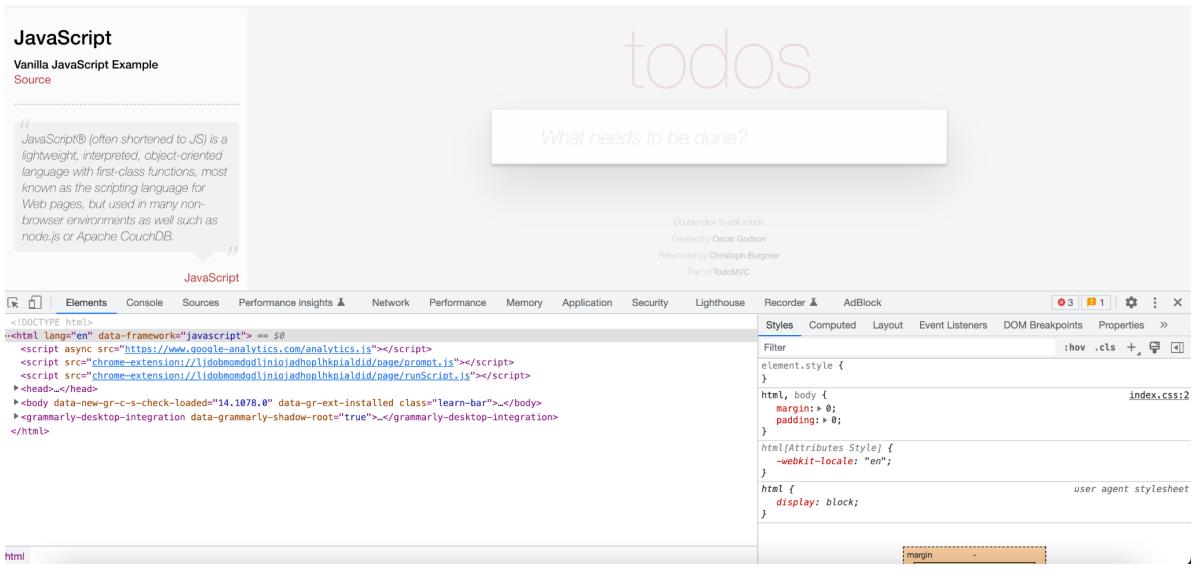
4. DevTools

- a. Right Click
- b. Inspect

C.



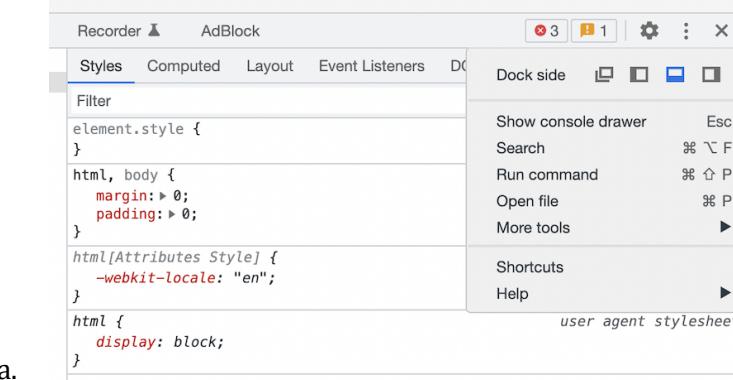
D.



i. DevTools allows us to

1. Have a look at the different elements of the page

5. Dock side:



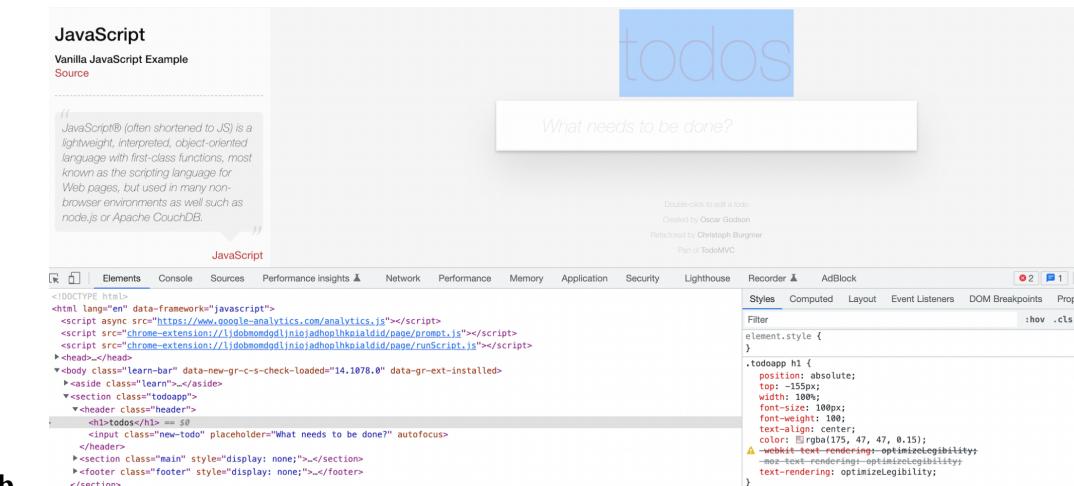
a.

6. Change the text of an element

a. Steps:

- i. Right-click on the element (Example - todos)
- ii. Inspect

1. DevTools will put the focus right into the area of the tree where my HTML is



b.

- i. Change the text from todos to DevTools
- ii. Hit ENTER
- iii. Text on the web page will also get changed.

JavaScript
Vanilla JavaScript Example
Source

```

<h1>DevTools</h1>

```

What needs to be done?

Double-click to edit a todo
Created by Oscar Godson
Reflected by Christoph Burgmer
Part of TodoMVC

Elements Console Sources Performance insights Network Performance Memory Application Security Lighthouse Recorder AdBlock

Styles Computed Layout Event Listener Filter element.style { } .todoapp h1 { position: absolute; top: -155px; width: 180px; font-size: 100px; text-align: center; color: #rgba(175, 47, 47, 0.15); -webkit-text-rendering: optimizeLegibility; -ms-text-rendering: optimizedLegibility; text-rendering: optimizeLegibility; }

1.

7. Change the styling of an element

- Under Styles (Bottom-Right corner)
- Font-size (100px to 200 px)

JavaScript
Vanilla JavaScript Example
Source

```

<h1>DevTools</h1>

```

What needs to be done?

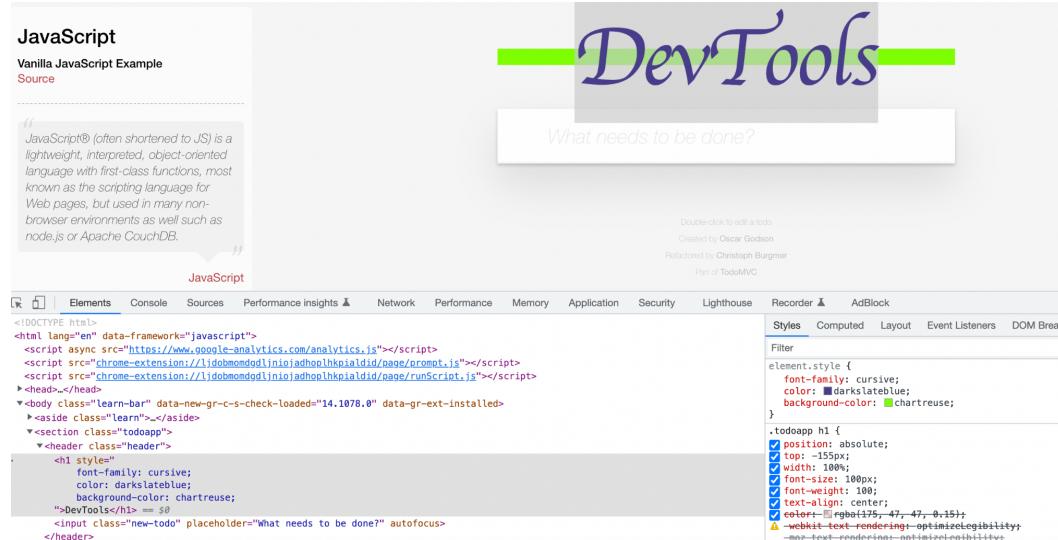
Double-click to edit a todo
Created by Oscar Godson
Reflected by Christoph Burgmer
Part of TodoMVC

Elements Console Sources Performance insights Network Performance Memory Application Security Lighthouse Recorder AdBlock

Styles Computed Layout Event Listener DOM Breakpoints Properties > Filter :hover .cls + ↻ element.style { } .todoapp h1 { position: absolute; top: -155px; width: 180px; font-size: 100px; text-align: center; color: #rgba(175, 47, 47, 0.15); -webkit-text-rendering: optimizeLegibility; -ms-text-rendering: optimizedLegibility; text-rendering: optimizeLegibility; }

1.

- Other properties added
- Font-family, color, background-color



2.

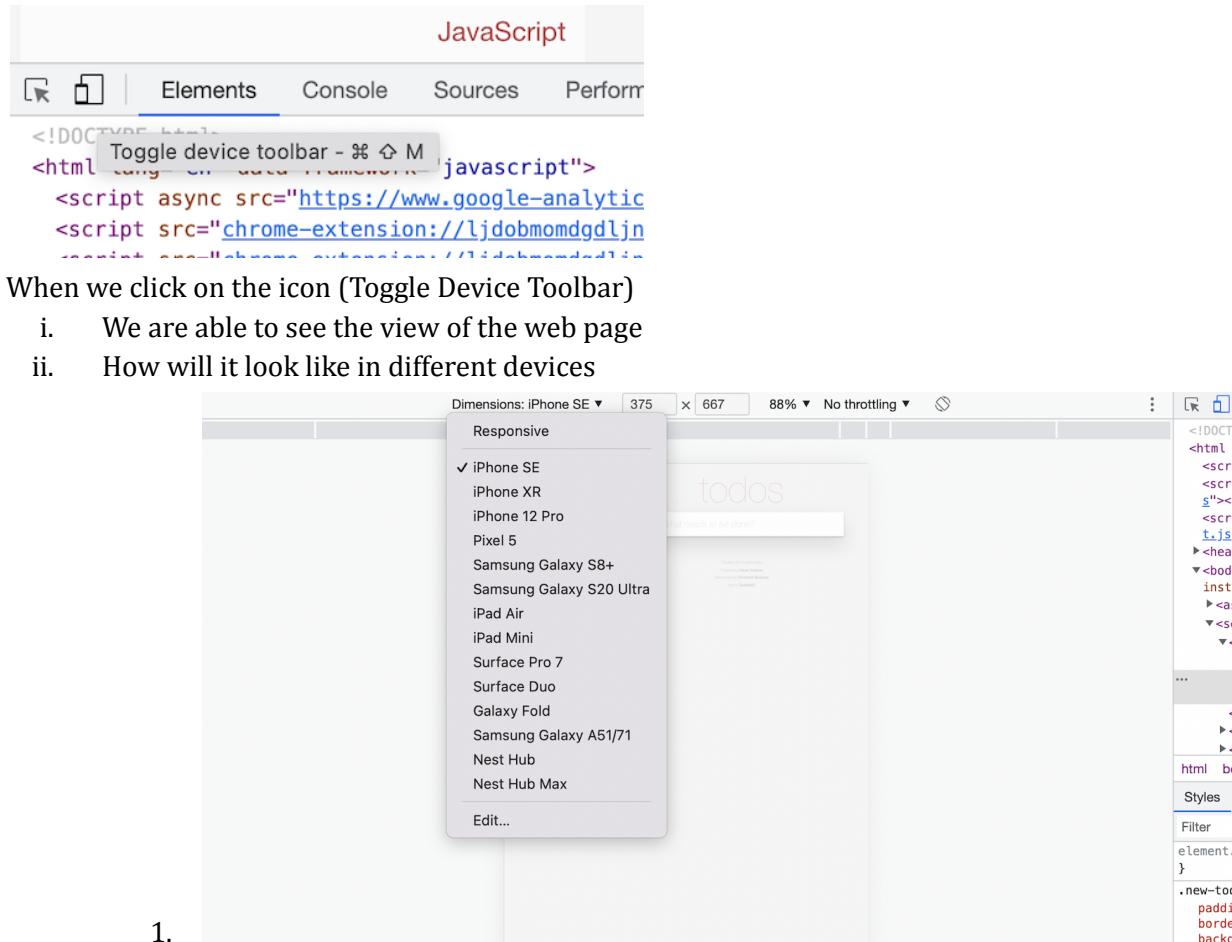
-----3. Elements Panel Exercise -----

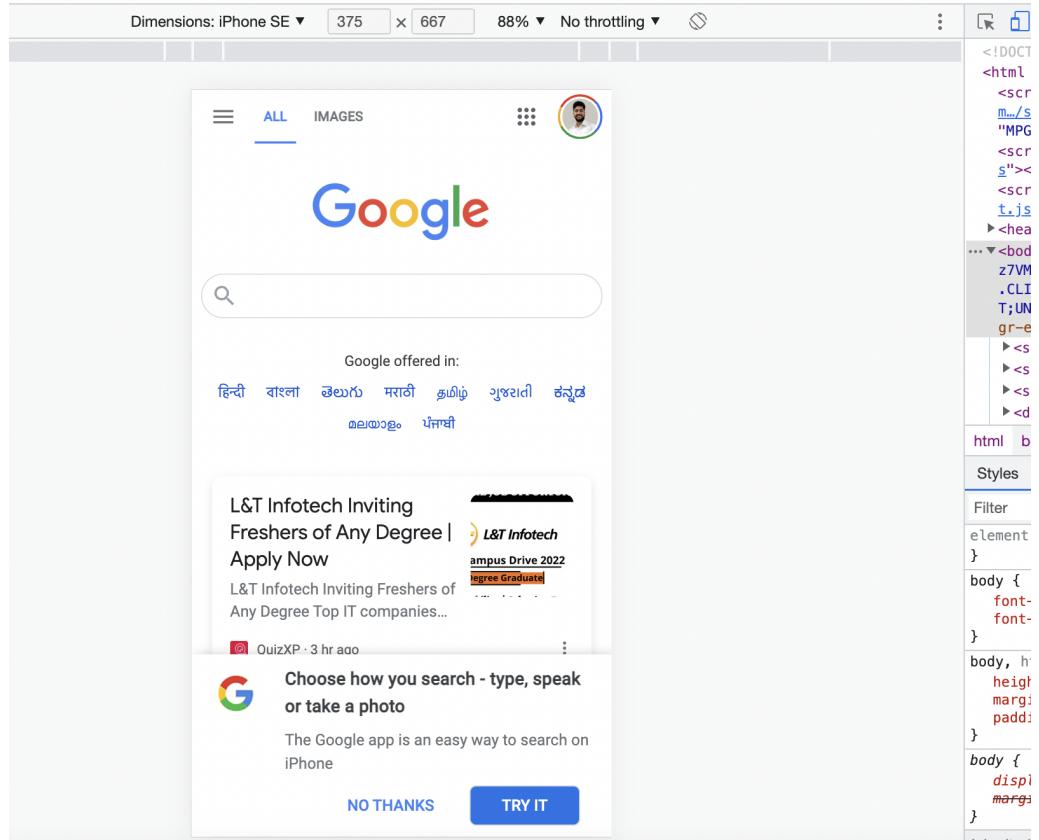
-----4. A Quick Tour of DevTools -----

1. Tabs:

- Elements
- Console
- Sources
- Performance Insights
- Network
- Performance
- Memory
- Application
- Security
- Lighthouse
- Recorder
- AdBlock

2. Toggle Device Toolbar:





1.

3. Console:

- a. We can write Javascript commands

```
Elements Console Sources Performance insights Network
top ▾ Filter
> console.log('UDEMY_Devtools Pro: The Basics of Chrome Developer Tools')
UDEMY_Devtools Pro: The Basics of Chrome Developer Tools
<- undefined
>
```

The screenshot shows the Chrome DevTools Console tab. It contains a single log entry: 'console.log('UDEMY_Devtools Pro: The Basics of Chrome Developer Tools')' followed by the resulting output 'UDEMY_Devtools Pro: The Basics of Chrome Developer Tools'. The 'Console' tab is highlighted.

b.

4. Sources:

- a. The source code of the application goes here

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar displays a tree view of files and projects. Under the 'examples/vanilla.js' project, the '(index)' file is currently selected. The main pane shows the code for 'injectedScript.js'. The code is a self-executing anonymous function that returns an object with a 'normalizeUrl' method. The 'normalizeUrl' method takes a URL as input and returns its hash. The code also includes a 'getReferrer' method that returns the normalized referrer URL.

```
1 'use strict'
2 ;(function (win) {
3   const exports = {}
4
5   ;(function (exports) {
6     const utils = {
7       /**
8        * Normalize URL
9        * @param {String} url
10       */
11      normalizeUrl(url) {
12        return this.hashUrl(url) || null
13      },
14
15      /**
16       * Get referrer.
17       */
18      getReferrer() {
19        return this.normalizeUrl(document.referrer)
20      }
21    }
22
23    exports.utils = utils
24  })(exports)
25
26  return exports
27})()
28
```

5. Application:

- a. This tab has the information regarding
 - i. Local Storage
 - ii. Session storage
 - iii. Cookies

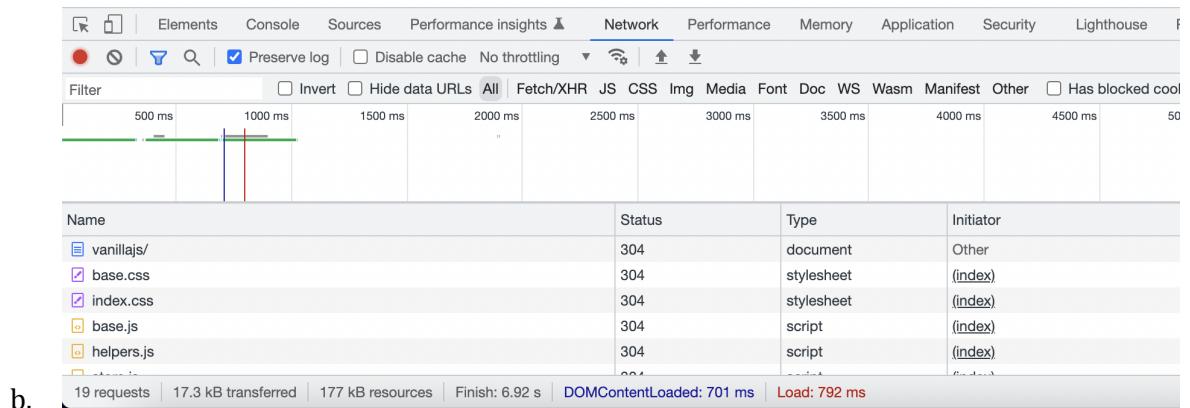
The screenshot shows the Chrome DevTools Network tab. On the left, the Storage section is expanded, showing Local Storage, Session Storage, IndexedDB, Web SQL, and Cookies. The Cookies section for the domain https://todomvc.com is selected, displaying two entries: '_gid' with value 'GA1.2.753630227.1662139290' and '_ga' with value 'GA1.2.1202128044.1662139290'. A modal dialog titled 'Select a cookie to preview its value' is open over the cookies table.

Name	Value	Domain	Path	Expires ...	Size
_gid	GA1.2.753630227.1662139290	.todomv...	/	2022-09...	30
_ga	GA1.2.1202128044.1662139290	.todomv...	/	2023-10...	30

Select a cookie to preview its value

6. Network:

- a. This panel shows all the Network Requests and Response details

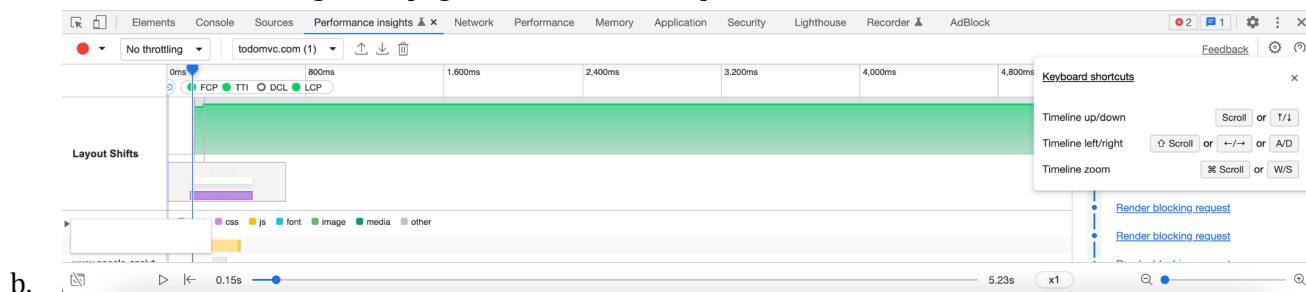


b.

7. Audit -> Performance Insights:

- a. This gives a bunch of tips

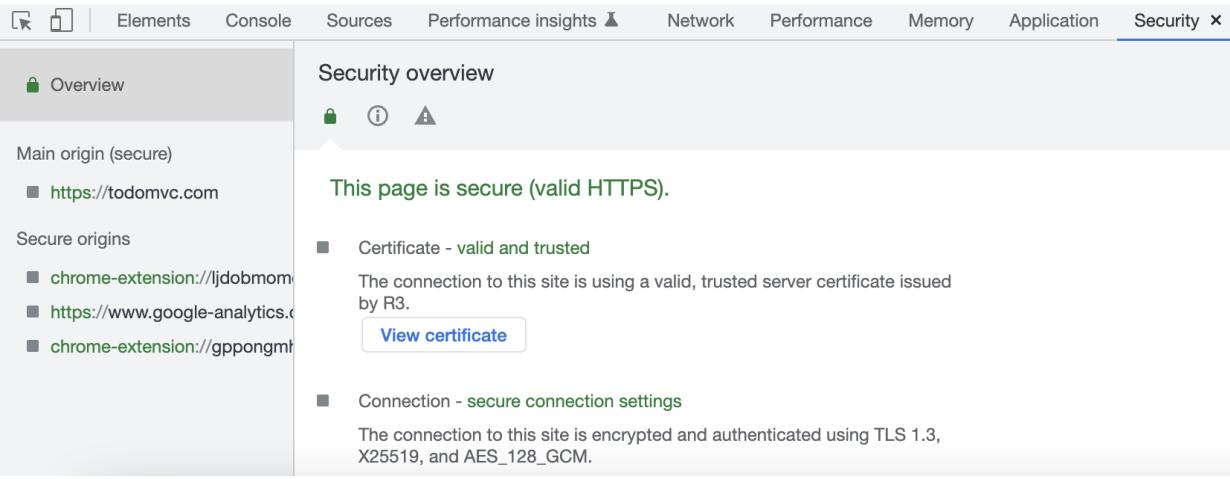
- i. How can we get our page to load a little bit quicker.



b.

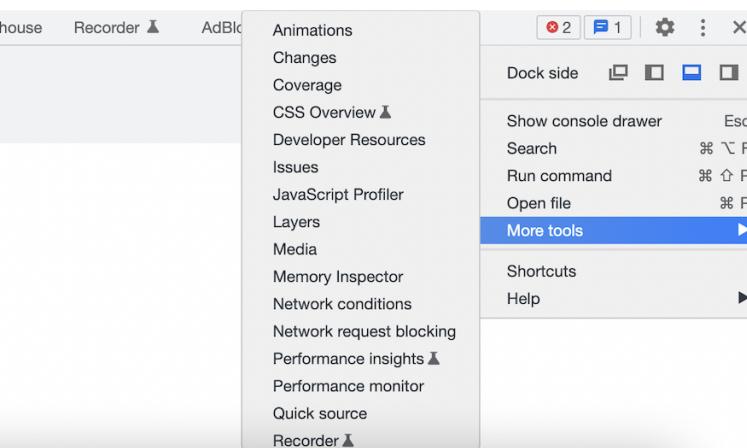
8. Security:

- This tab gives information regarding the Security of the web page.



The screenshot shows the 'Security' tab in the Chrome DevTools interface. The main content area is titled 'Security overview' and displays a green lock icon indicating a secure connection. Below this, a message states 'This page is secure (valid HTTPS)'. Two sections are listed: 'Certificate - valid and trusted' and 'Connection - secure connection settings'. The 'View certificate' button is visible under the certificate section. On the left sidebar, under 'Main origin (secure)', the URL 'https://todomvc.com' is shown. Under 'Secure origins', several URLs are listed, including 'chrome-extension://ljdobmmonjgklnfjgkhljgkldfjgk', 'https://www.google-analytics.com', and 'chrome-extension://gppongmfpnjkfjgklnfjgkldfjgk'. The letter 'b.' is positioned to the left of the screenshot.

9. More Tools:



The screenshot shows the 'More tools' menu in the Chrome DevTools interface. The menu is organized into several sections: Animations, Changes, Coverage, CSS Overview, Developer Resources, Issues, JavaScript Profiler, Layers, Media, Memory Inspector, Network conditions, Network request blocking, Performance insights, Performance monitor, Quick source, and Recorder. A right-click context menu is open over the 'More tools' section, listing options like 'Show console drawer' (Esc), 'Search' (⌘ ⌂ F), 'Run command' (⌘ ⌄ P), 'Open file' (⌘ P), and 'More tools' (highlighted with a blue background). Other options include 'Shortcuts' and 'Help'. The letter 'a.' is positioned to the left of the screenshot.

-----5. Using the Console to Run Javascript -----

1. Tab -> Console:

- To get all the p tags present on the page
 - document.getElementsByTagName('p')

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. On the left, there's a preview of a web page with several `p` tags. In the console, the command `document.getElementsByTagName('p')` is run, and it returns a `HTMLCollection` object with 5 items. The output is:

```
> document.getElementsByTagName('p')
<HTMLCollection(5) {p, p, p, p, p}
> 0: p
> 1: p
> 2: p
> 3: p
> 4: p
> length: 5
> [[Prototype]]: HTMLCollection
```

ii.

-----6. Exercise: Color Changes with Javascript -----

-----7. Sources Tab and Basic Debugging -----

- This is helpful to check the application behavior/flow

- How exactly our code is working

The screenshot shows the Chrome DevTools 'Sources' tab. The left sidebar shows a file tree with `js` and `node_modules` folders. The `view.js` file is selected. The right pane shows the code for `view.js`:

```
1 /*global qs, qsa, $on, $parent, $delegate */
2
3 (function (window) {
4     'use strict';
5
6     /**
7      * View that abstracts away the browser's DOM completely.
8      * It has two simple entry points:
9      *
10     * - bind(eventName, handler)
11     *   Takes a todo application event and registers the handler
12     * - render(command, parameterObject)
13     *   Renders the given command with the options
14     */
15     function View(template) {
16         this.template = template;
17
18         this.ENTER_KEY = 13;
+this.ESCAPE_KEY = 27;
```

b.

- We can put the Breakpoints

- Perform a click on the line number

-----8. Snippets: The Console on Steroids -----

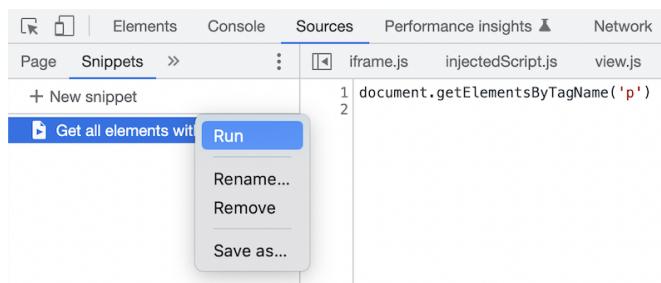
1. This is helpful to save a snippet of code for usage in future

- a. Sources -> Snippets

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. In the left sidebar, under the 'Snippets' section, there is a snippet titled 'Get all elements with p tag'. The code content area contains the following:

```
1 document.getElementsByTagName('p')
```

b.



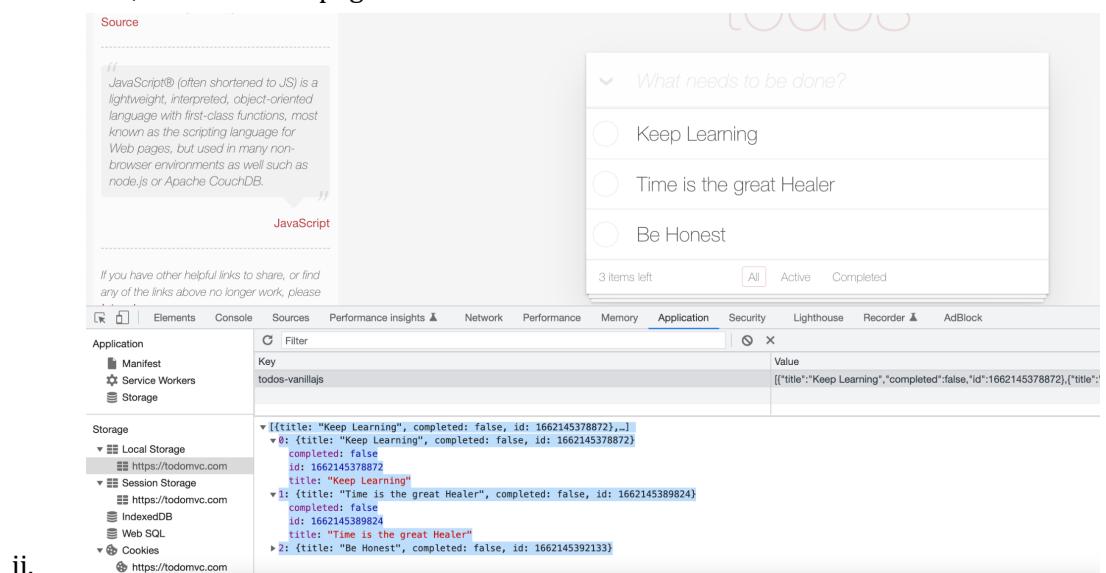
c.

=====Section 2: Digging Deeper=====

-----9. Cookies and Local Storage with the Application Tab -----

1. Tab - Application:

- I added some items in Todo App
 - And, refreshed the page



The screenshot shows the Chrome DevTools Application tab open. In the left sidebar, under 'Storage', the 'Local Storage' section is selected, showing items for the domain 'https://todomvc.com'. Three items are listed:

Key	Value
todos-vanillajs	[{"title": "Keep Learning", "completed": false, "id": 1662145378872}, {"title": "Time is the great Healer", "completed": false, "id": 1662145389824}, {"title": "Be Honest", "completed": false, "id": 1662145392133}]

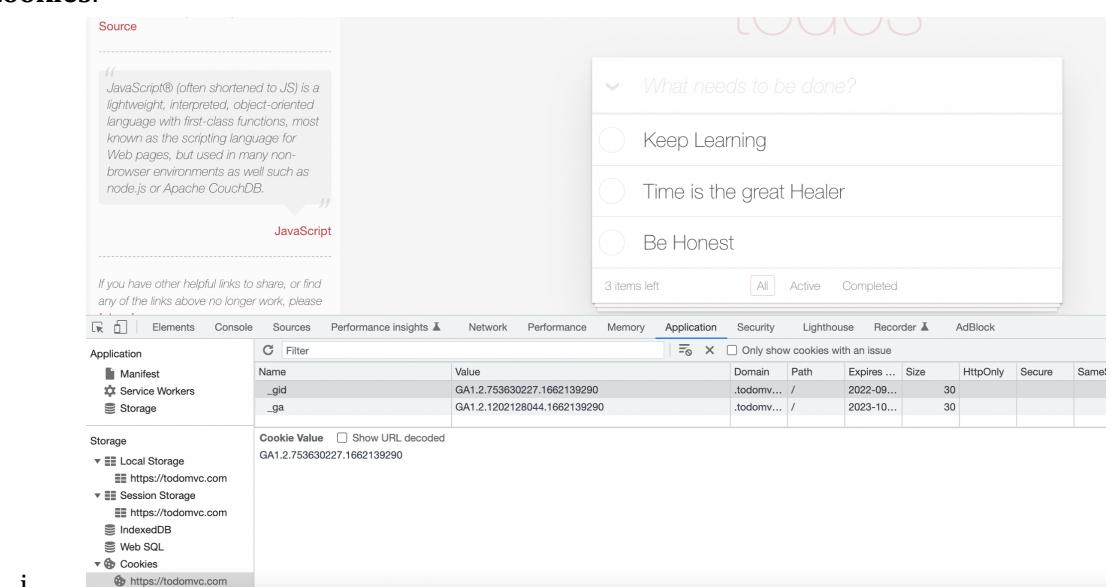
On the right, the main pane displays the 'To Do' application interface with three items: 'Keep Learning', 'Time is the great Healer', and 'Be Honest'.

1. Still, Able to see the items added

a. Why?

i. This application is using **LocalStorage**

b. Cookies:



The screenshot shows the Chrome DevTools Application tab open. In the left sidebar, under 'Storage', the 'Cookies' section is selected, showing items for the domain 'https://todomvc.com'. Two cookies are listed:

Name	Value	Domain	Path	Expires ...	Size	HttpOnly	Secure	SameSite
_gid	GA1.2.753630227.1662139290	.todomv...	/	2022-09...	30			
_ga	GA1.2.1202128044.1662139290	.todomv...	/	2023-10...	30			

On the right, the main pane displays the 'To Do' application interface with three items: 'Keep Learning', 'Time is the great Healer', and 'Be Honest'.

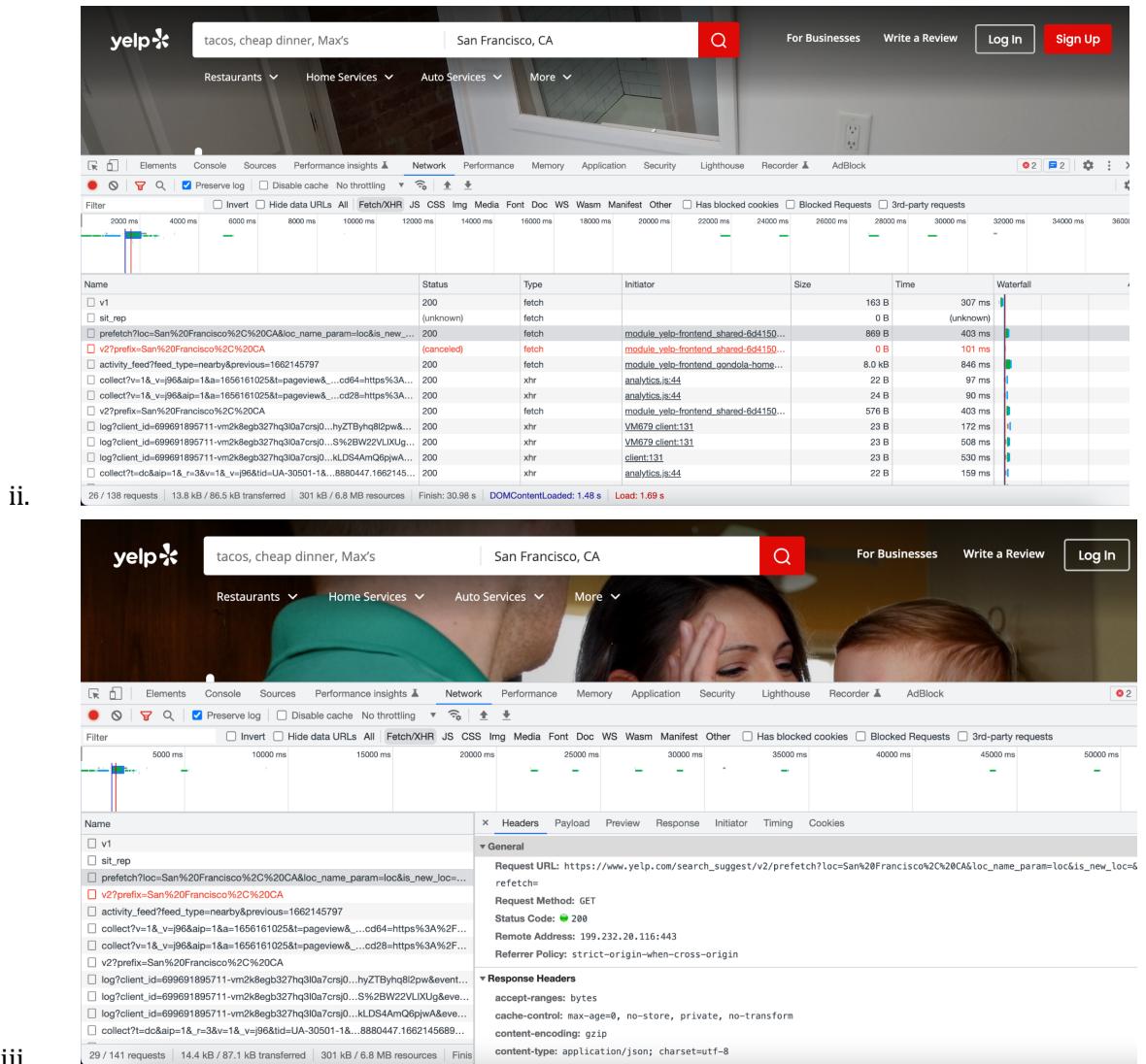
i.

-----10. Local Storage Exercise -----

-----11. A Peak at Network Calls -----

1. Tab - Network:

- <https://www.yelp.com/>
- DevTools -> Network
- Click on Fetch/XHR (Default - All is selected)



-----12. Network Calls Exercise -----

-----13. Page Load Speed with Audits (Old) -> Performance insights -----

1. Performance insights:

JavaScript
Vanilla JavaScript Example
Source

JavaScript® (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, most known as the scripting language for Web pages, but used in many non-browser environments as well such as node.js or Apache CouchDB.

JavaScript

Elements Console Sources Performance insights Network Performance Memory Application Security Lighthouse Recorder AdBlock

No throttling

Get actionable insights on your website's performance

- Record or import a trace into the Performance Insights panel
- Get an overview of your page's runtime performance
- Identify improvements to your performance via a list of actionable insights

Measure page load Import from file...

a.

i. Click on the Measure page load

1.

-----14. Performance, Memory, and Security -----

1. Tab - Performance:

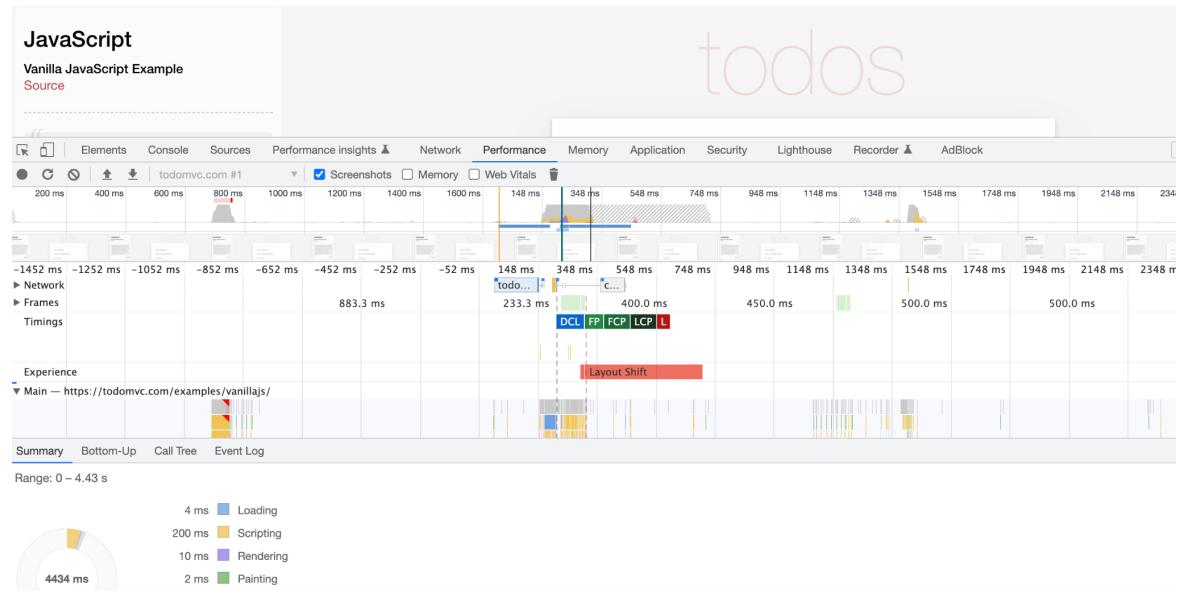
Click the record button or hit ⌘ E to start a new recording.

Click the reload button or hit ⌘ ⌘ E to record the page load.

After recording, select an area of interest in the overview by dragging. Then, zoom and pan the timeline with the mousewheel or WASD keys. [Learn more](#)

a.

i. We can record all the things that are happening on the page when we load.



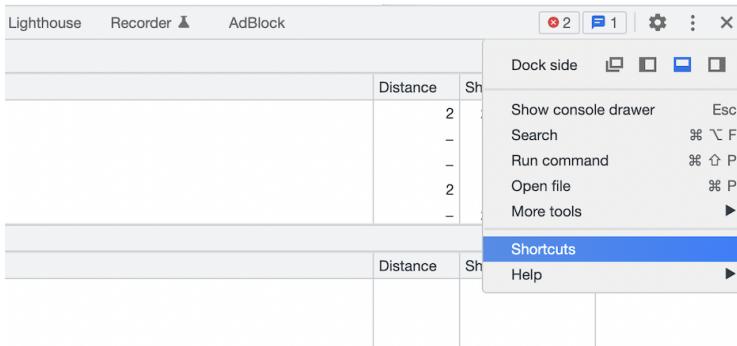
2. Tab - Memory:

a.

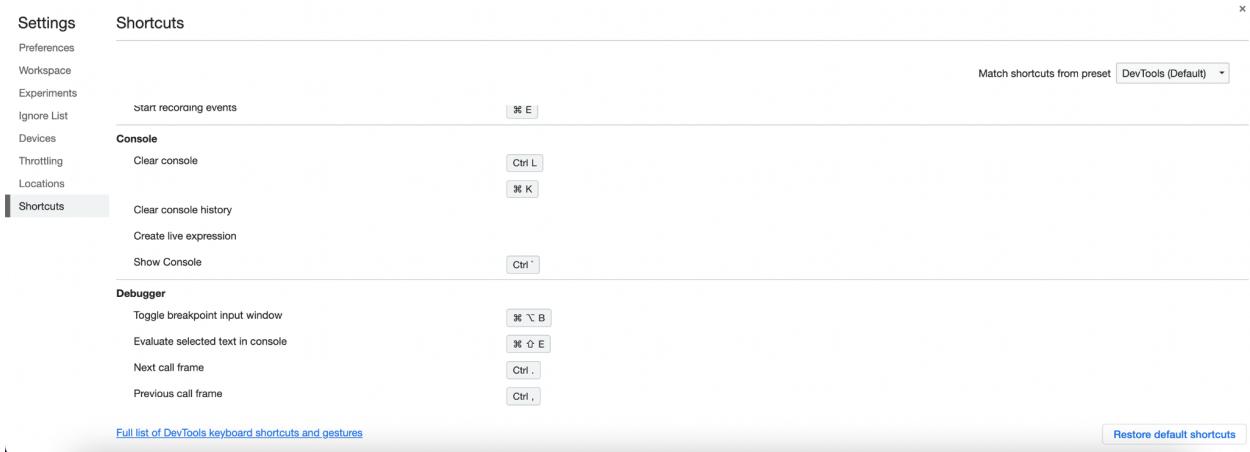
	Distance	Shallow Size	Retained Size
Constructor	2	2 785 912 22 %	7 068 768 55 %
▶ (system) <x12697	-	1 215 744 10 %	5 166 316 40 %
▶ (closure) <x0914	-	1 696 0 %	4 689 384 37 %
HEAP SNAPSHTOS	2	1 937 232 15 %	3 891 756 30 %
▶ Snapshot 1 12.8 MB	-	2 758 780 22 %	3 741 840 29 %
Profiles			
▶ Detached HTMLInputElement <x5			
▶ (array) <x3939			
▶ (compiled code) <x61517			
Retainers			
Object			

b.

-----15. The Most Useful Keyboard Shortcuts -----



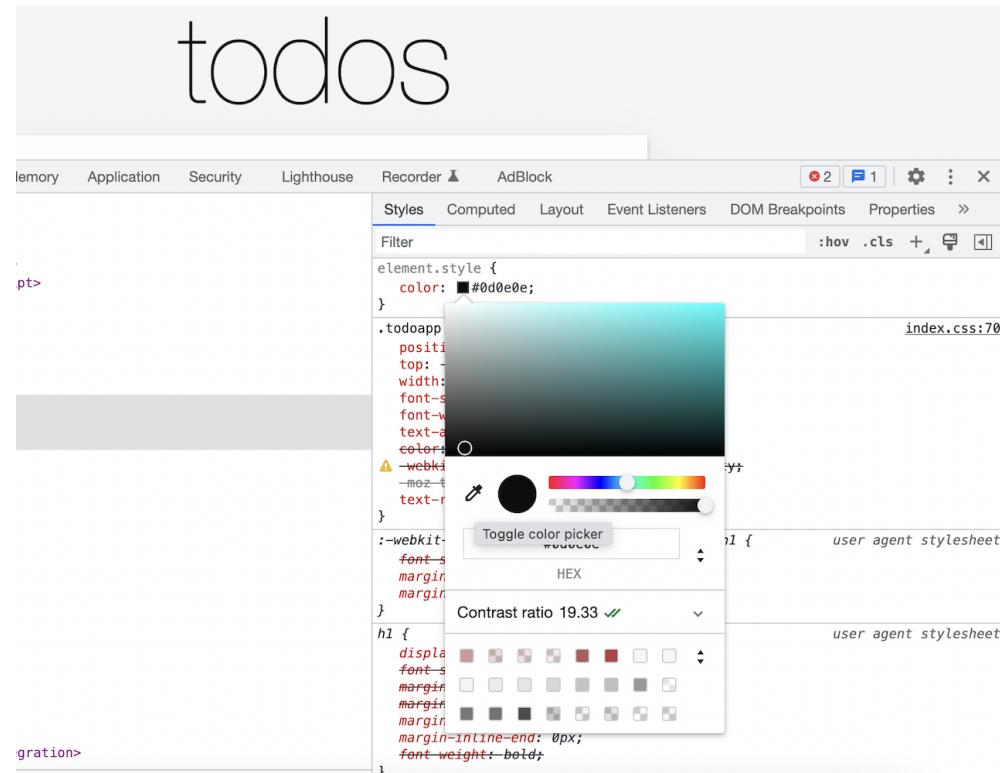
1.



2.

-----16. Styling Editors -----

1. Choose the color from any element present on the page
 - a. Using icon - Toggle color picker
 - i. Inspect **Element - A**
 - ii. Elements -> Styles
 1. Under element.style
 - a. Add color
 - iii. Click on this icon (Toggle color picker)
 - iv. Click on **Element - B** (of which element you want to use the color)
 - v. The color of **Element - B** will be used for **Element - A**



2. Also, we can use the
 - a. Sliders
 - b. Hex - RGBA format

-----17. Styling Editors Exercise -----

=====Section 3: Bonus=====

-----18. Bonus! Coupon at 1:15! Other Resources & My Twitter -----

1. More documents:

- a. <https://github.com/rajatt95/Documents>

2. To connect:

- a. <https://www.linkedin.com/in/rajat-v-3b0685128/>
- b. <https://github.com/rajatt95>
- c. <https://rajatt95.github.io/>

THANK YOU!

