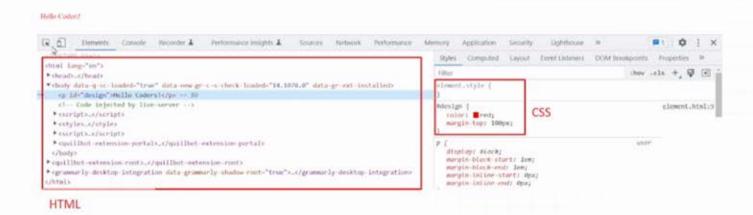
JAVASCRIPT DEBUG, WEB DEBUG, DEV TOOL MASTERING



Element Tab Overview

Element tab is mainly used to debug the code of **HTML** and **CSS** in your website. Whenever we need to change something in the styling of our HTML and CSS and we want to see how it will reflect in our website, we can do that with the help of this tab.



< 2 > :

Element Tab Overview

Style - shows you all of the rules that apply to an element, including declarations that have been overridden.

Computed - displays the final set of CSS properties the browsen is applying to the selected element. It also displays a representation of an element's box model, so you can quickly examine and edit its content, padding, border and margin areas.

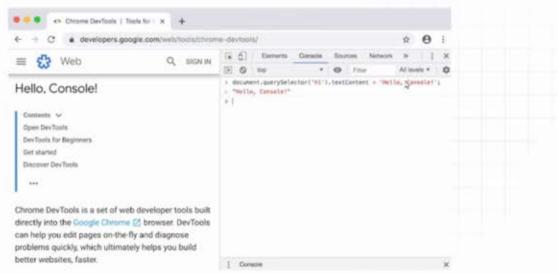
Event Listeners - see all the event listeners bound to the element.

DOM breakpoints - DOM Breakpoints are useful to break on DOM mutation events, such as when a node is removed, modified, or its attributes are changed.

Accessibility - The Accessibility pane is where you can view the accessibility tree, ARIA attributes, and computed accessibility properties of DOM nodes.

Console Tab Overview

Console tab helps us in debugging the JavaScript code. We can view the logged message and run the JavaScript code here. This tab helps us in showing the error along with the line number in our source code



Console API Overview

console.log(): Prints a message to the Console

console.clear(): Clears the console.

console.count(): Writes the number of times that count() has been invoked

console.countReset(): resets the count

console.error: Prints object to the Console, formats it as an error, and includes a stack trace

b.

console.table(): Logs an array of objects as a table.

More about console api: https://developer.chrome.com/docs/devtools/console/api/

Console Utilities API Overview

clear(): clears the console of its history.

copy(object): copies a string representation of the specified object to the clipboard.

debug(function): When the specified function is called, the debugger is invoked and breaks inside the function on the **Sources** panel allowing to step through the code and debug it.

getEventListeners(object): returns the event listeners registered on the specified object.

monitor(function): When the function specified is called, a message is logged to the console that indicates the function name along with the arguments that are passed to the function when it was called.

monitorEvents(object [, events]): When one of the specified events occurs on the specified object, the Event object is logged to the console.

D

More about console utilities api: https://developer.chrome.com/docs/devtools/conso

Sources Tab Overview

Sources tab allows you to set breakpoints and evaluate expressions in Javascript.

```
Elements Console
                                           Performance insights &
                              Recorder A
                                                                     Sources Network Performance Memory Application Security
                                     [4] my6 is * content is
      Hillesystem Overrides 20
☐ ▲ Group files by Authored/Deployed
                                       14) const loadSingleUser = () => (
                                             fetch("https://randomuser.me/api/")
Send feedback
                                              then((res) => res. (son())
Learn More
                                               .then((data) -> comsole.log(data.results(8)));
r □ top
                                       147 3:

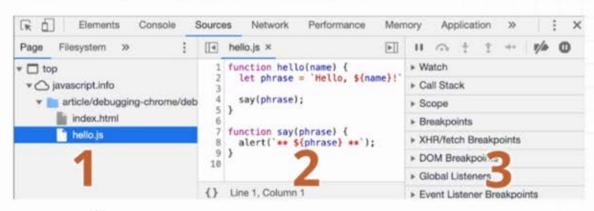
◆ △ 127.0.0.1:5500

                                       149 loadSingleUser();
     layouthini
                                       151 const items = document.getElementsDvClassName("buttons");
     my6.js
                                       15J for (const item of items)
 ➤ Contailwindess.com
                                             item.add[ventListener("click", function (event) (
                                               const result - confirm("Are you sure?");
                                               console.log(result);
                                       256
                                              if (result on true) (
                                                 event.target.parentNode.remove(hild(event.target);
                                       1516
                                       159
                                             1)1
                                       358
                                       162 window.addfventiistener("Jusd", function gen() (
                                             let number = Math.round(Math.random() * 10000);
                                             if (number.to5tring().length (- 4) (
                                       165
                                               gen();
                                     ()
                                                                                                                                         Cover
Console What's New X Issues
```

Sources Tab Overview

Sources tab allows you to set breakpoints and evaluate expressions in Javascript.

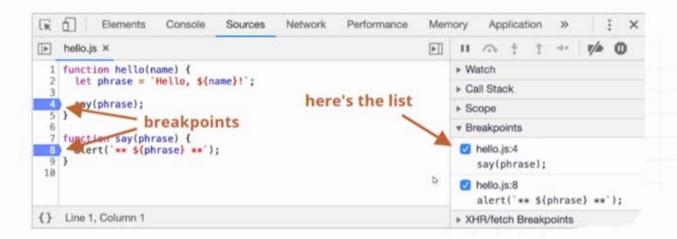
- The File Navigator pane lists HTML, JavaScript, CSS and other files, including images that are attached to the page.
- 2. The Code Editor pane shows the source code.
- The JavaScript Debugging pane is for debugging





Sources Tab (Breakpoints)

A breakpoint is a point of code where the debugger will automatically pause the JavaScript execution.

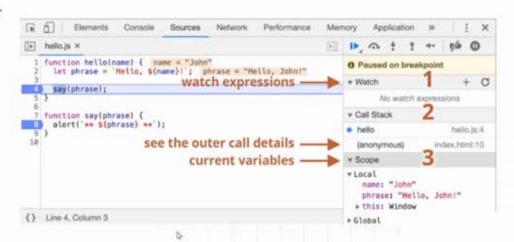


Sources Tab (Watch, Call Stack, Scope)

- Watch shows current values for any expressions.
- Call Stack shows the nested calls chain.
- Scope current variables.

Local shows local function variables.

Global has global variables (out of any functions).



SLIDE

