

[Open in app](#)[Get started](#)

Published in Frontend Weekly · [Follow](#)



Amnah khatun · [Follow](#)

Oct 14, 2021 · 3 min read



Debug like a pro

Being a web developer one must be aware of some awesome debugging techniques. To be honest I have been using `console.log()` along with debugger to debug my code, but debugging is not limited to these two. I have come across some great articles informing there is so much more. Now I use combination of many techniques which I have explained below, and believe me my life has become easier 😊

Let's start with showing message in the console

- `log()`
- `error()`
- `warn()`
- `table()`
- `count()`
- `time()` and `timeEnd()`



[Open in app](#)[Get started](#)

```
< undefined
> console.error("Displaying an error")
✖ ▶ Displaying an error
< undefined
> console.warn("Warning")
⚠ ▶ Warning
< undefined
> console.table({"firstName": "John", "lastName": "Doe"})

  (index)      Value
  firstName    "John"
  lastName     "Doe"
  ▶ Object
< undefined
> for(let i=0;i<5;i++){
  console.count(i);
}
0: 1
1: 1
2: 1
3: 1
4: 1
< undefined
> console.time('abc');
  let fun1= function(){
    console.log('fun is running');
  }
  let fun2 = function(){
    console.log('fun2 is running..');
  }
  fun1();
  fun2();
  console.timeEnd('abc');
fun is running
fun2 is running..
abc: 0.3740234375 ms
```

Few keyboard shortcuts that come handy

To clear console **ctrl L**(Windows) or **cmd K**(Mac OS)





Open in app

Get started

While logging multiple variables, keeping track of individual variables can become difficult. Either we have to write them with different `console.log()` or give them specific strings.



Join 30K+ Developers and Stay on Top of Frontend Development

Get frontend related articles, links and tutorials right in your inbox. 8 links/week only.
No fluff. No spam.

Sign up



I agree to leave medium.com and submit this information, which will be collected and used according to [Upscribe's privacy policy](#).

Using object literal comes in handy here



[Open in app](#)[Get started](#)

```
console.log(a,"a");  
console.log(b,"b");
```

```
10 "a"
```

```
20 "b"
```

```
< undefined
```

```
> let a = 10;  
   let b = 20;  
   //using object literal  
   console.log({a},{b});
```

```
▶ {a: 10} ▶ {b: 20}
```

Few Developer tools settings which I personally use

CSS Overview — This will show the summary of all the colors, fonts, media queries all in one place.





Open in app

Get started

Overview summary

Colors

Font info

Unused declarations

Media queries

Overview summary

Elements

121

External stylesheets

1

Inline style elements

34

Style rules

389

Media queries

5

Type selectors

14

ID selectors

11

Class selectors

287

Universal selectors

0

Attribute selectors

0

Non-simple selectors

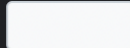
55

Colors

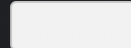
Background colors: 3



#FFFFFF

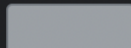


#F8F9FA

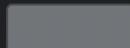


#F2F2F2

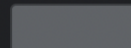
Text colors: 8



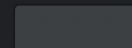
#9AA0A6



#70757A



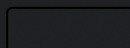
#5F6368



#3C4043



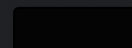
#1A0DAB



#202124



#000000



#000000DE

- On Google Chrome, open *Developer Tools*.
- Open **Settings**.
- Click on **Experiments** Tab.
- Enable the **CSS Overview**.
- Reload the **DevTools**.



[Open in app](#)[Get started](#)

The screenshot shows the Chrome DevTools interface. The top bar includes tabs for Elements, Console, CSS Overview, Sources, and Network. The Elements panel displays a tree view of the document structure, with a 'flex' class highlighted. The CSS Overview panel shows the 'Font Family' dropdown and 'CSS Properties' section with sliders for Font Size, Line Height, Font Weight, and Spacing. The Styles pane shows the 'element.style' and '.css-1s44ra' rules.

```
... 5lkjjo div.css-g1v4tt header.css-17t02fm div.css-...
Styles Computed Layout Event Listeners DOM
Filter
element.style {
}
.css-1s44ra, [data-css-1s44ra] {
  padding-top: 15px;
  text-align: center;
  font-size: 24px;
  letter-spacing: 0.01em;
  font-weight: 200;
}
*, :after, :before {
  box-sizing: inherit;
}
* {
  margin: 0;
  padding: 0;
}
```

- Go to **Experiments** Tab through Chrome DevTools.
- Check **Enable New Font Editor Tools** within **Styles** Pane.
- Reload the **DevTools**.
- Select the **HTML** element, which includes the font properties you want to change.



[Open in app](#)[Get started](#)

Few shortcuts which I always use

Get currently selected tag

Go to chrome dev tools and type `$0` in the console. This will give the reference to the currently selected node.

The image shows the React website on the left and the Chrome DevTools console on the right. The website has a header with the React logo and a navigation bar. The console shows the DOM tree with the `<h1 class="css-159p4b7">React</h1>` element selected. The console output shows the result of typing `$0`, which is `<h1 class="css-159p4b7">React</h1>`.

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting

Get the last evaluated value in the console

Go to chrome dev tools and type `$_` in the console. This will always print the last evaluated value.

The image shows the Chrome DevTools console with several examples of the `$_` shortcut. The first example shows `2 + 3` being evaluated to `5`. The second example shows `$_ + 5` being evaluated to `10`. The third example shows `$_ * 10` being evaluated to `100`.

Copy request as cURL from the network tab

Directly copy the request as a cURL from the network tab



[Open in app](#)[Get started](#)

Filter ☐ Hide data URLs

All **Fetch/XHR** JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies

☐ Blocked Requests

20 ms 40 ms 60 ms 80 ms 100 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
app-data.json	304	xhr	loader.js:41	70 B	110 ...	
app-data.json				75 B	97 ms	
page-data.js			ref...		16 ms	
page-data.js				83 B	98 ms	
blog				62 B	115 ...	
app-data.json						

Open in new tab

Clear browser cache

Clear browser cookies

Copy

Block request URL

Block request domain

Replay XHR

Sort By

Header Options

Save all as HAR with content

Copy link address

Copy response

Copy stack trace

Copy as fetch

Copy as Node.js fetch

Copy as cURL

Copy all as fetch

Copy all as Node.js fetch

Copy all as cURL

Copy all as HAR

Edit whole page by turning designMode to on

Go to chrome dev tools and type **document.designMode="on"**. This will let you modify without refreshing the page.

I hope this has helped you as it has helped me.

In case if you know other tricks or shortcuts and want to help the community, do let me know in the comment section 🙌.

Happy coding.





Open in app

Get started

Get an email whenever Amnah khatun publishes.

<https://medium.com/subscribe/@amnah.k>

Your email

Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

