



Debugging Tools

or how to leave console.log in prod code

Frontend Junior Program - 2021

Achievement despite ignorance



Getting it to Work

And Having No Idea How

ONLY?

@ThePracticalDev

Agenda

- 1 Intro
- 2 Debugging Tools
- 3 Chrome, Edge DevTools
- 4 Fiddler, Charles proxy
- 5 Nightly browser, extensions
- 6 Summary

INTRODUCTION

Preface

Working with JavaScript (even in projects), you **will have** this feeling:

Ahh, it is clear and easy!

ten minutes later ...

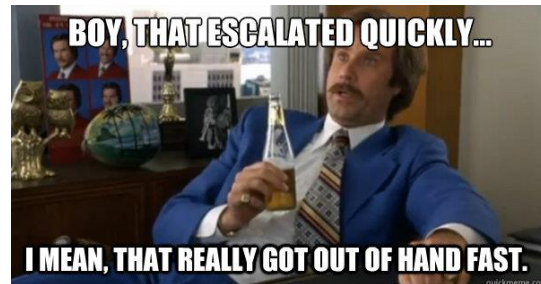
What the hell is going on?

A terrible mistake you can do at that point is that you try to make it work *somehow*, with several trial and error attempts.

At best, it won't work, and you just spent a lot of time on it.

At worst, it will work, and you *certainly introduced a bug* – and sometimes that will be very hard to find.

But don't worry, you don't need to do something wrong, bugs just born in their own rights.



"Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."

The rubber duck method

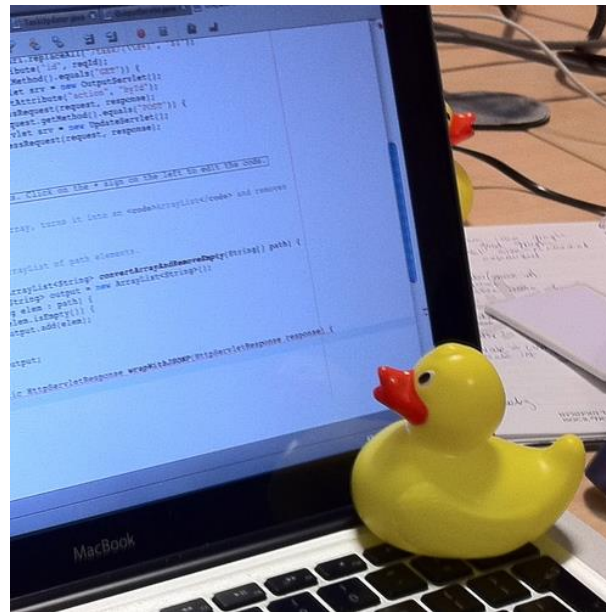
If you face with an issue, just slow down and try to fix it **step by step** - with understanding every detail of it.

Use the [rubber duck method](#) – it works!

Your colleagues also could play the role of the rubber duck. They often share tips and tricks, and it really helps to build relationship and think in teamwork.

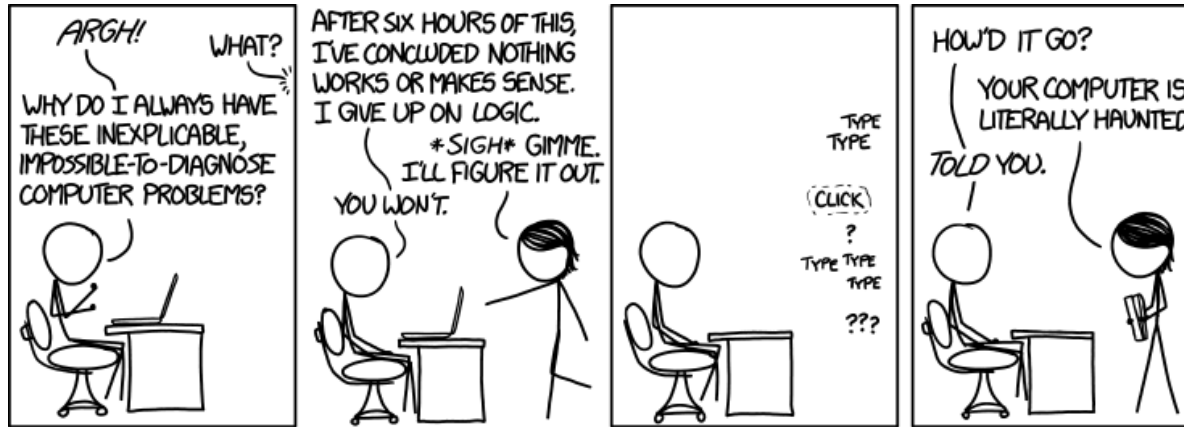
Also, it could really help to relieve your [impostor syndrome](#).

While **unit testing** could help you to reason about the code, when the bundle reaches the browser, **debugging tools** will do the heavyweight in finding them.



this duck is a way stronger than you may think: it forces you to go through **step by step** (otherwise, you could skip seemingly trivial parts again and again)

What is debugging?



Debugging – is a process of search and elimination of errors in a code and providing information on the state of a system.

Debugging allows you to ...

1

check the call stack

4

override the data

2

view the data

5

monitor variables

3

see the errors

6

add breakpoints

DEBUGGING TOOLS

Browser dev tools



Edge / Chrome

Built-in debugger based
on Chromium



Firefox

Built-in debugger



Safari

Built in debugger
iOS debugger

Proxies

Charles is an HTTP proxy / HTTP monitor / Reverse Proxy that enables a developer to view all of the HTTP and SSL / HTTPS traffic between their machine and the Internet. This includes [requests](#), [responses](#) and the [HTTP headers](#) (which contain the cookies and caching information).

In Web and Internet development you are unable to see what is being sent and received between your web browser / client and the server. Without this visibility it is difficult and time-consuming to determine exactly where the fault is. Charles makes it easy to see what is happening, so you can quickly diagnose and fix problems.

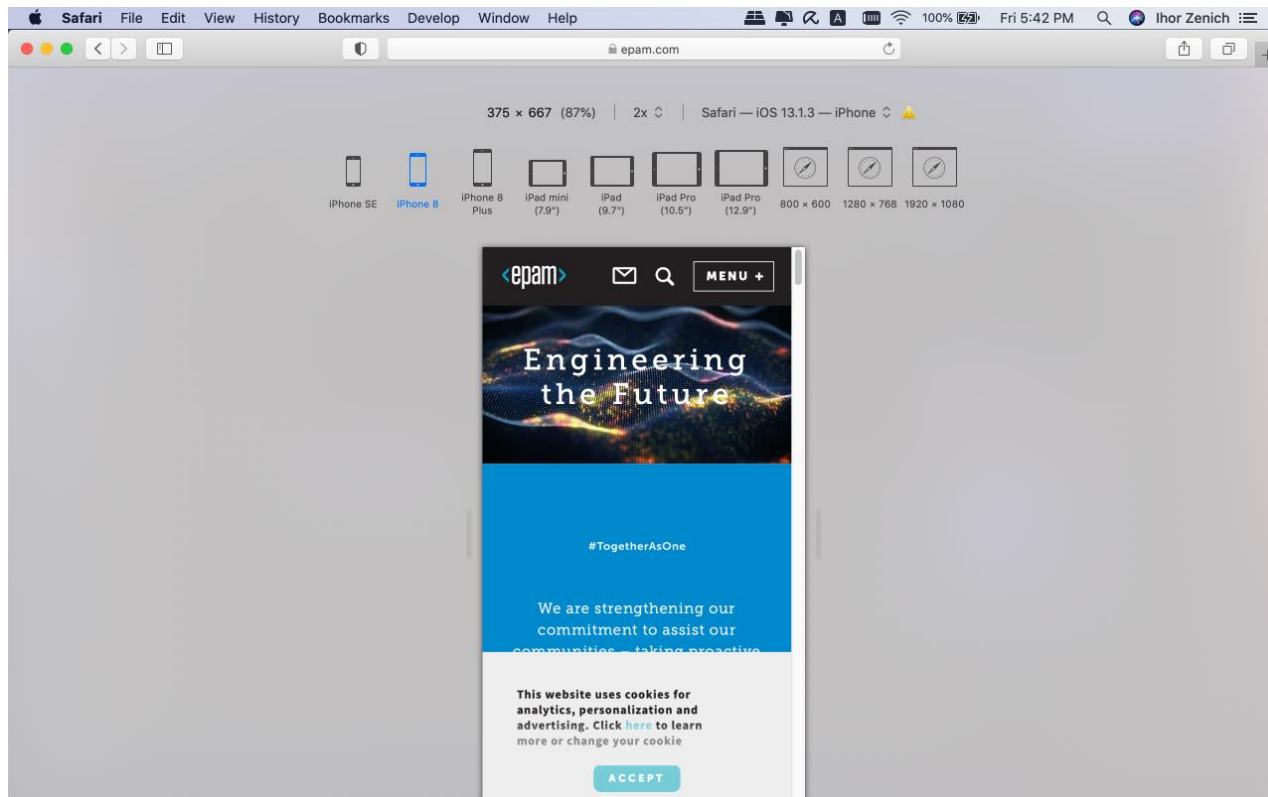


Charles proxy



Fiddler

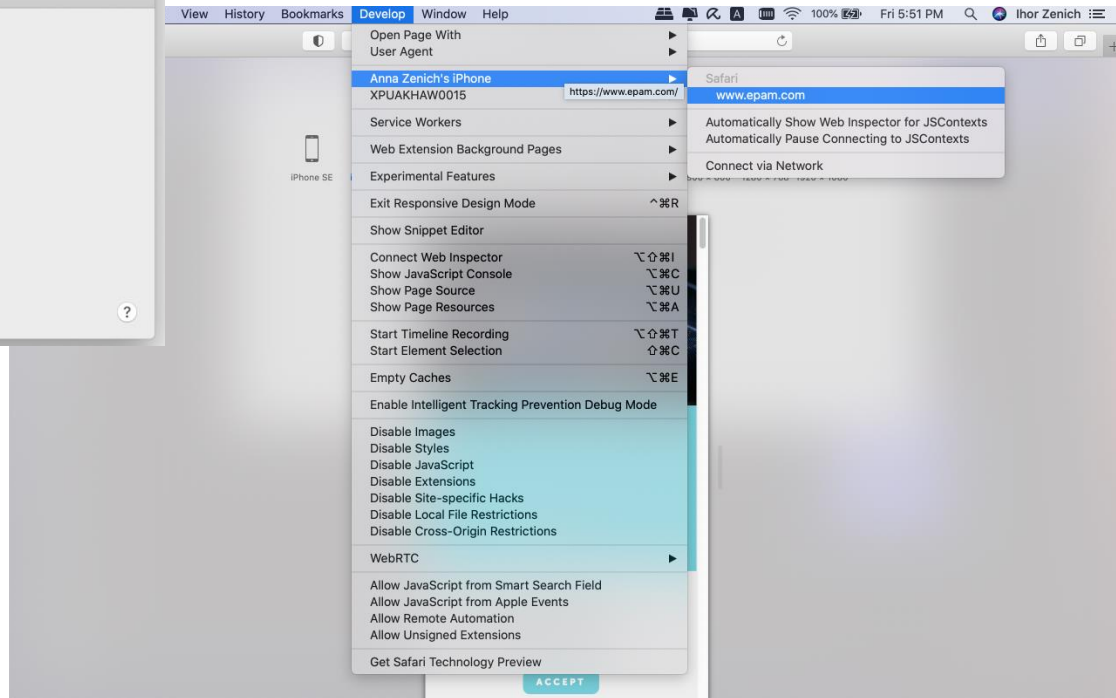
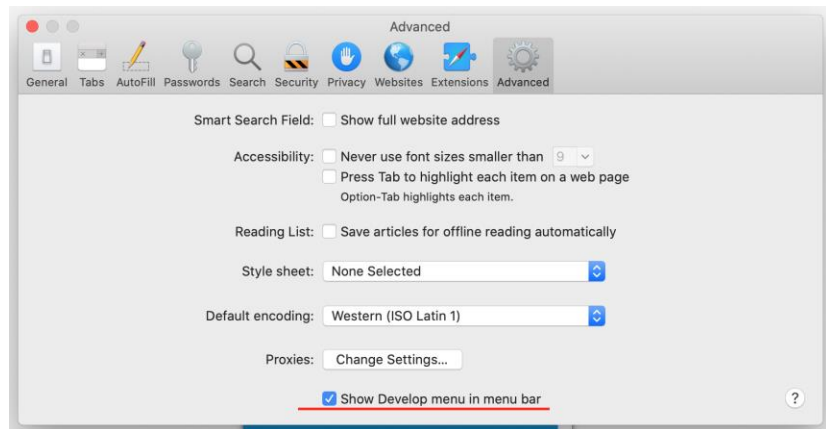
Safari Developer tools



iOS devices remote debugging



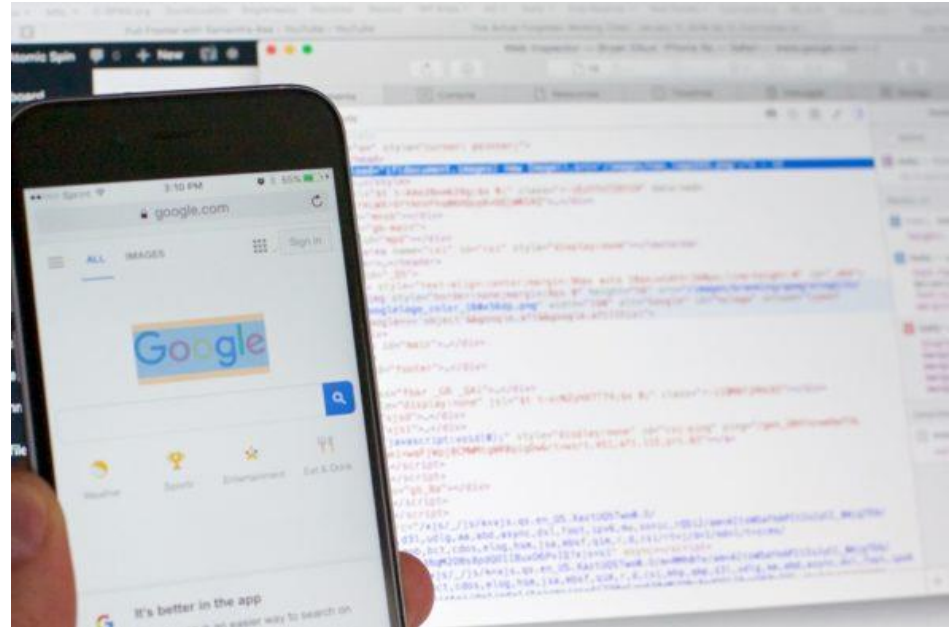
Safari – preferences (⌘,) and connecting a device



iOS devices remote debugging



After you connected an iOS device with Safari, remote debugging makes it possible to load a website on the device and [inspect in the desktop Safari](#).



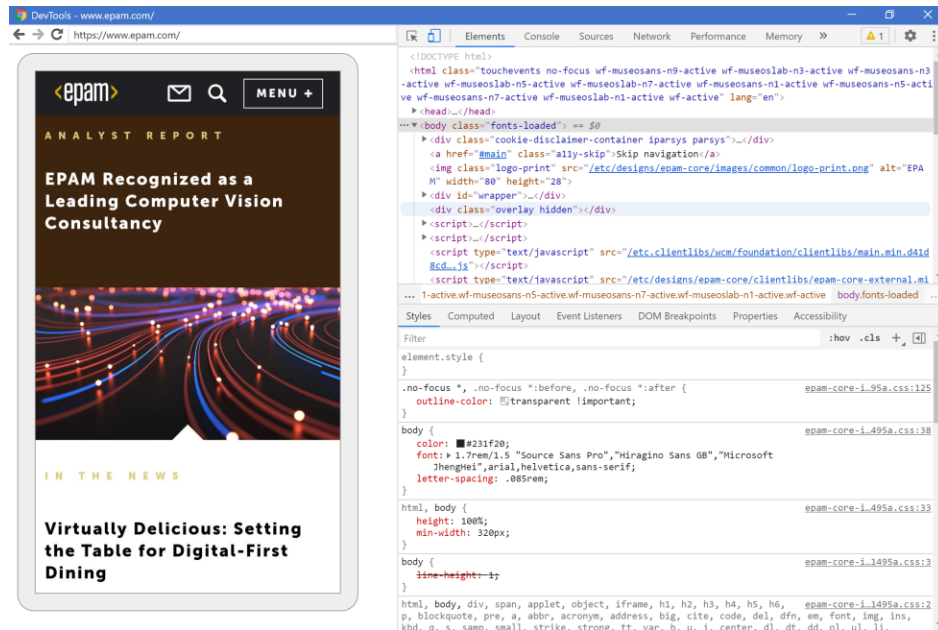
Android devices remote debugging



The same is possible with Android devices: both the desktop Chrome and Edge are capable to connect to an Android Chrome – and it works on Windows and on macOS as well.

Steps

1. connect the device with a [USB cable](#)
2. enable [remote debugging](#) on the phone
3. open [chrome://inspect](#) in the desktop browser (edge://inspect)
4. open Chrome on the Android device



Firefox Developer Tools



The screenshot displays the Firefox Developer Tools interface over a web browser showing the EPAM website. The browser's address bar shows the URL `https://www.epam.com/our-work/travel-and-hospitality`. The website header includes the EPAM logo and navigation links: SERVICES, HOW WE DO IT, OUR WORK, INSIGHTS, ABOUT, CAREERS, CONTACT US, and Global (EN). The main content area features a large image of two people with luggage, and text indicating the company was established in 2002 and recognized as a top 10 travel and hospitality provider.

The Developer Tools interface is open, showing the following panels:

- Inspector:** Displays the HTML structure of the page. The selected element is a `<div class="section_wrapper section--padding-large">` element.
- Style Editor:** Shows the CSS styles for the selected element. The styles include `max-width: 100%; padding-left: 0px; padding-right: 0px; padding: 10.2rem 2rem; padding-right: 2rem; padding-left: 2rem;`.
- Layout:** Shows the box model for the selected element. The dimensions are 1523x204. The box model properties are: `border-box`, `display: block`, `float: none`, `line-height: 25.5px`, and `position: relative`.

CHROME / EDGE DEVTOOLS



chrome
DEVELOPER TOOLS

[Chrome DevTools](#)

Main panels



Elements

Inspect and edit
markup and styles



Network

Page-load performance



Sources

Debug with breakpoints



Application

Manage your local data



Lighthouse

Analyse a page as it loads



Console

Interact from command Line

Elements

DevTools - www.w3schools.com/css/css3_animations.asp

Elements Console Sources Network Performance Memory Application Security Audits

```
<!doctype html>
<html lang="en-US" style="height: 100%;" class="mdl-js">
  <head>...</head>
  <body style="position: relative; min-height: 100%; top: 0px;">
    <div class="w3-container top"> == $0
      ::before ← Pseudo element
        <a class="w3schools-logo" href="//www.w3schools.com">...</a>
        <div class="w3-right w3-hide-small w3-wide toptext" style="font-family: 'Segoe UI', Arial, sans-serif">THE WORLD'S LARGEST WEB DEVELOPER SITE</div>
        ::after
      </div>
      <div style="display:none;position:absolute;z-index:4;right:52px; height:44px;background-color:#5f5f5f;letter-spacing:normal;" id="googleSearch">...</div>
      <div style="display:none;position:absolute;z-index:3;right:111px; height:44px;background-color:#5f5f5f;text-align:right;padding-top:9px;" id="google_translate_element">...</div>
      <div class="w3-card-2 topnav" id="topnav" style="position: relative;">...</div>
      <div class="w3-sidebar w3-collapse" id="sidnav" style="top: 112px;">...</div>
    </body>
  </html>
```

html.mdl-js body div.w3-container.top

Styles Computed Event Listeners

Filter Toggle element state (pseudo class) Add class

Force element state

- ☐ :active
- ☐ :focus
- ☐ :focus-within
- ☐ :hover
- ☐ :visited

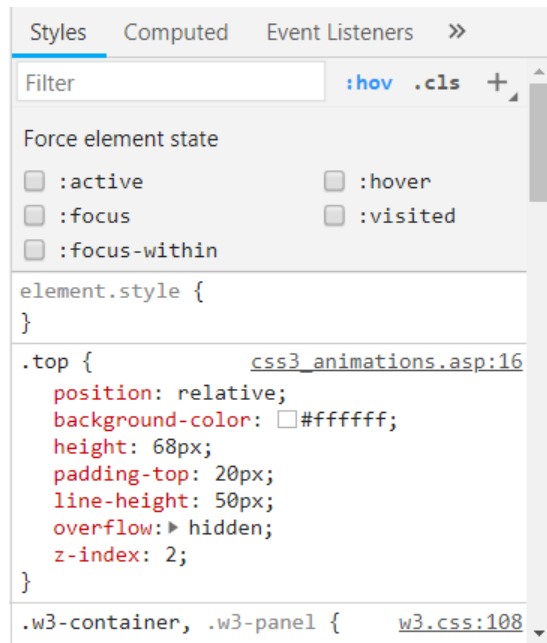
element.style { }

.top {
 position: relative;
 background-color: #ffffff;
 height: 68px;
 padding-top: 20px;
 line-height: 50px;
 overflow: hidden;
 z-index: 2;
}

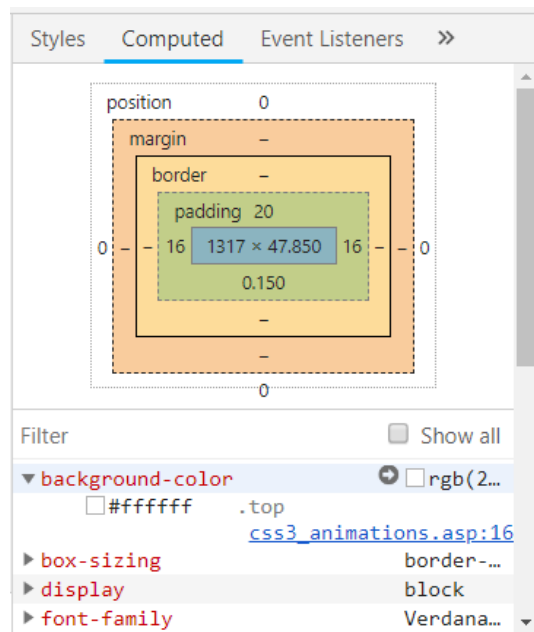
.w3-container, .w3-panel {

Elements – Styles, Computed

Styles

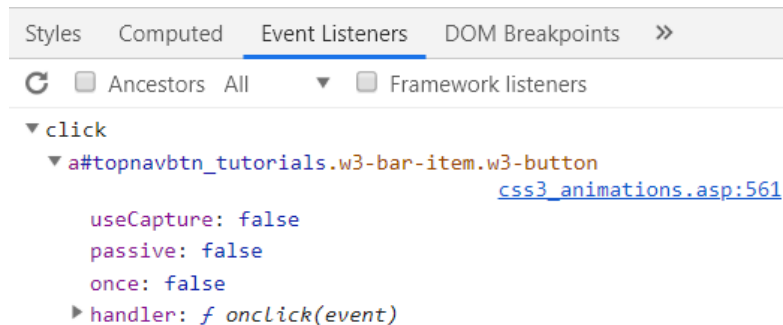


Computed



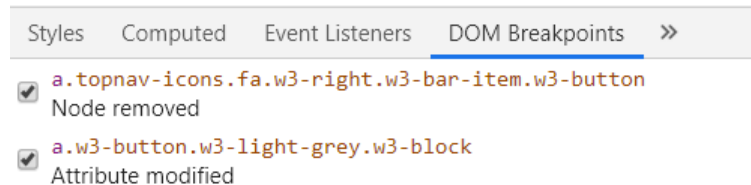
Elements - Event Listeners, DOM Breakpoints

Event Listeners



The screenshot shows the 'Event Listeners' tab selected in the browser's developer tools. The top bar includes 'Styles', 'Computed', 'Event Listeners' (active), and 'DOM Breakpoints'. Below the bar, there are checkboxes for 'Ancestors' and 'All', and a dropdown menu. The main area shows a tree view of event listeners. Under the 'click' event, a listener is listed for the element `a#topnavbtn_tutorials.w3-bar-item.w3-button` with the URL `css3_animations.asp:561`. The listener details show `useCapture: false`, `passive: false`, `once: false`, and a handler `handler: f onClick(event)`.

DOM Breakpoints



The screenshot shows the 'DOM Breakpoints' tab selected in the browser's developer tools. The top bar includes 'Styles', 'Computed', 'Event Listeners', and 'DOM Breakpoints' (active). Below the bar, there are checkboxes for 'Ancestors' and 'All', and a dropdown menu. The main area shows a list of DOM breakpoints. Two breakpoints are listed: `a.topnav-icons.fa.w3-right.w3-bar-item.w3-button` with the event 'Node removed', and `a.w3-button.w3-light-grey.w3-block` with the event 'Attribute modified'.

Viewport emulation

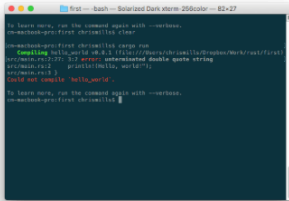
iPhone 6/7/... 414 x 736 79% DPR: 3.0 Mobile

▼ Debugging isn't scary

When writing code of some kind, everything is usually fine, until that dreaded moment when an error occurs — you've done something wrong, so your code doesn't work — either not at all, or not quite how you wanted it to. For example, the following shows an error reported when trying to compile a simple program written in the `Rust` language.

```
1 $ ls -la src
2 total 16
3 -rw-rw-r-- 1 user user 4096 Jan 10 10:10 src.rs
4
5 $ rustc src.rs
6 error: unterminated double quote string
7   --> src.rs:5:17
8    |
9    | println!(Hello, world!);
10   |                ^^^^^^
11
12 $ ls -la src
13 total 16
14 -rw-rw-r-- 1 user user 4096 Jan 10 10:10 src.rs
15
16 $ rustc src.rs
17 error: unterminated double quote string
18   --> src.rs:5:17
19    |
20    | println!(Hello, world!);
21    |                ^^^^^^
22
23 $
```

Here, the error message is relatively easy to understand — “unterminated double quote string”. If you look at the listing, you can probably see how `println!(Hello, world!);` might logically be missing a double quote. However, error messages can



Elements Console Sources Network Performance

<!doctype html>

<html lang="en" dir="ltr" class>

<head prefix="og: http://ogp.me/ns#"></head>

<body data-slug="Learn/HTML/Introduction_to_HTML/Debugging_HTML" contextmenu="edit-history-menu" data-search-url class="document">

<script></script>

<ul id="nav-access">

<!-- Header -->

<header id="main-header" class="header-main"></header>

<!-- Content will go here -->

<main id="content" role="main"> == \$0

<!-- heading -->

<div id="wiki-document-head" class="document-head">...

</div>

<div class="center clear"></div>

</main>

<div id="developer_needs" class="mdn-cta-container hidden" aria-expanded="false" aria-hidden="true" data-banner="developer_needs"></div>

<!-- Footer -->

<footer id="nav-footer" class="nav-footer"></footer>

<!-- site js -->

<!--[if lte IE 8]><script type="text/javascript" src="/static/build/js/selectivizr.8bb9e662e963.js" charset="utf-8"></script><![endif]>>

<script src="/static/js/i18n/en-US/javascript.414b87adc480.js"></script>

<script type="text/javascript" src="/static/build/js/main.b0f831aaadf5.js" charset="utf-8"></script>

<div class="notification-tray" role="status" aria-live="polite"></div>

html body.document main:content

Console

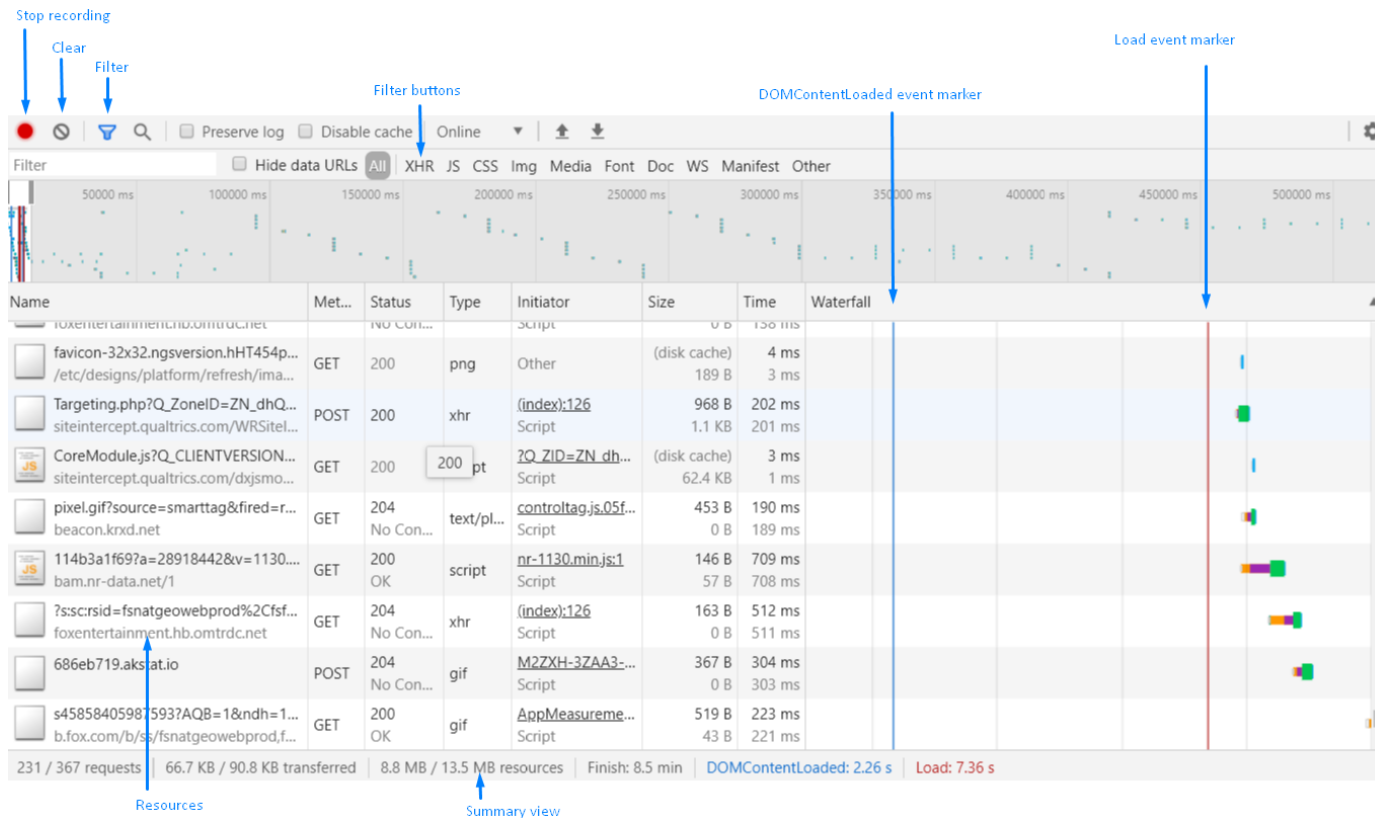
top Filter

epam

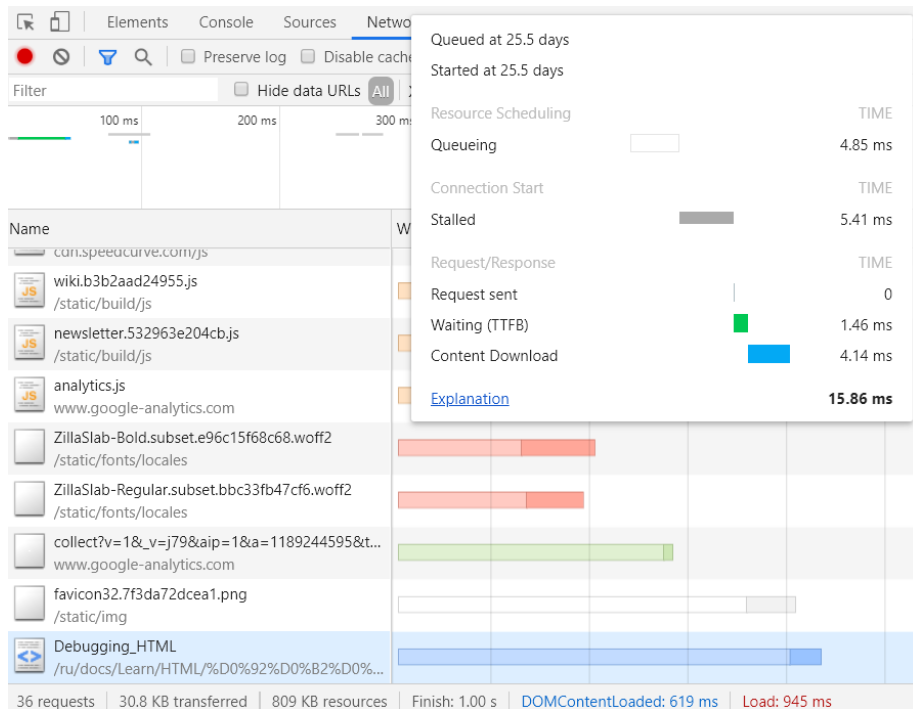
CONFIDENTIAL | © 2022 EPAM Systems, Inc.

24

Network



Network - data visualization



- Timing details visualization
- Load event markers
- Different sorting options:
Start Time, Duration, Latency etc.

Network - Headers

The screenshot shows the Network tab of a web browser's developer tools. The left pane lists network requests, with 'javascript.d69af44e1987.js' selected. The right pane shows the 'Headers' tab for this request. The 'General' section displays request details: URL, method (GET), status (200), remote address, and referrer policy. The 'Response Headers' section lists various headers including access-control, cache-control, content-encoding, content-type, date, last-modified, server, status, vary, via, and x-amz-cf-id.

Elements Console Sources **Network** Performance Memory Application Security Audits Adblock 2

Filter ☐ Hide data URLs ☒ All XHR JS CSS Img Media Font Doc WS Manifest Other

Name

- analytics.js
- javascript.d69af44e1987.js /static/jsi18n/ru
- main.b0f831aaddf5.js /static/build/js
- wiki.b3b2aad24955.js /static/build/js
- newsletter.532963e204cb.js /static/build/js
- web-docs-sprite.22a6a085cf14.svg /static/img
- arrow-left.52215ddfcd4.svg /static/arrows
- arrow-up.7bcf357a852e.svg /static/arrows
- arrow-right.cbc8b4f075cc.svg /static/arrows
- file.7ac510b78865.svg /static/file-icons
- chevron-right.a102b6151ae9.svg

36 requests | 30.8 KB transferred | 809 KB resources

Headers Preview Response Timing

General

- Request URL:** https://developer.mozilla.org/static/jsi18n/ru/javascript.d69af44e1987.js
- Request Method:** GET
- Status Code:** 200 (from memory cache)
- Remote Address:** 13.32.100.252:443
- Referrer Policy:** no-referrer-when-downgrade

Response Headers

- access-control-allow-origin:** *
- age:** 5347450
- cache-control:** max-age=315360000, public, immutable
- content-encoding:** gzip
- content-type:** application/javascript; charset="utf-8"
- date:** Tue, 16 Jul 2019 18:04:22 GMT
- last-modified:** Mon, 15 Jul 2019 17:24:45 GMT
- server:** meinheld/0.6.1
- status:** 200
- vary:** Accept-Encoding
- via:** 1.1 2f43ce8da1e384926701eb877d7472ba.cloudfront.net (CloudFront)
- x-amz-cf-id:** HTU72b-6nPWKx7jqexbapGHY2EztTUKYsviMiZ8f2qtzFr3ajjDCA==

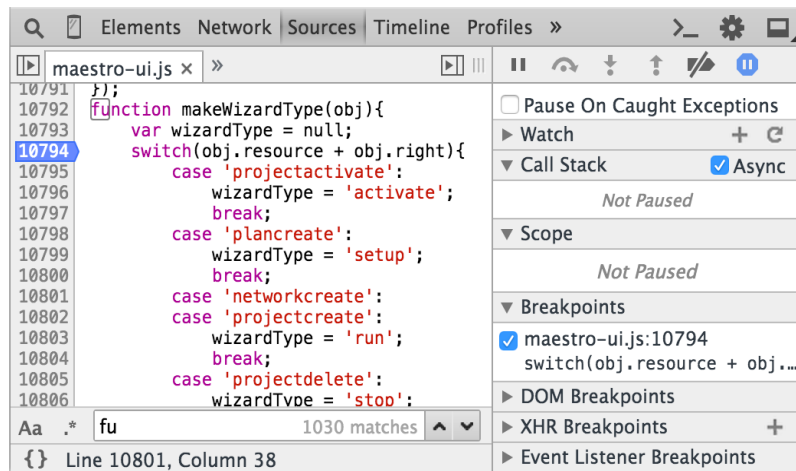
- Headers and payload
- Response body
- Cookies
- Timing

Sources - Breakpoints

Old School*

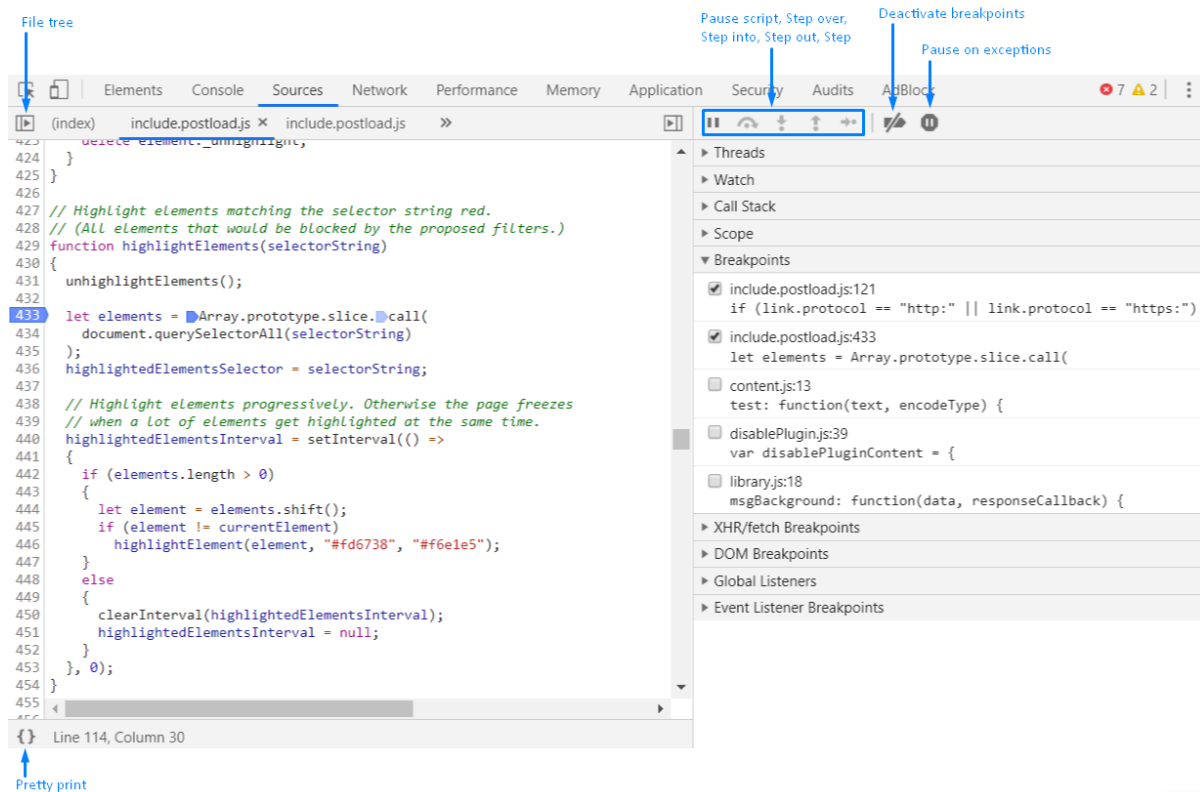
```
function makeWizardType(obj) {  
  var wizardType = null;  
  alert(obj.resource + obj.right);  
  switch (obj.resource + obj.right) {  
    case 'projectactivate':  
      wizardType = 'activate';  
      break;  
    case 'plancreate':  
      wizardType = 'setup';  
      break;  
    case 'networkcreate':  
    case 'projectcreate':  
      wizardType = 'run';  
      break;  
    case 'projectdelete':  
      wizardType = 'stop';  
      break;  
  }  
}
```

DevTools

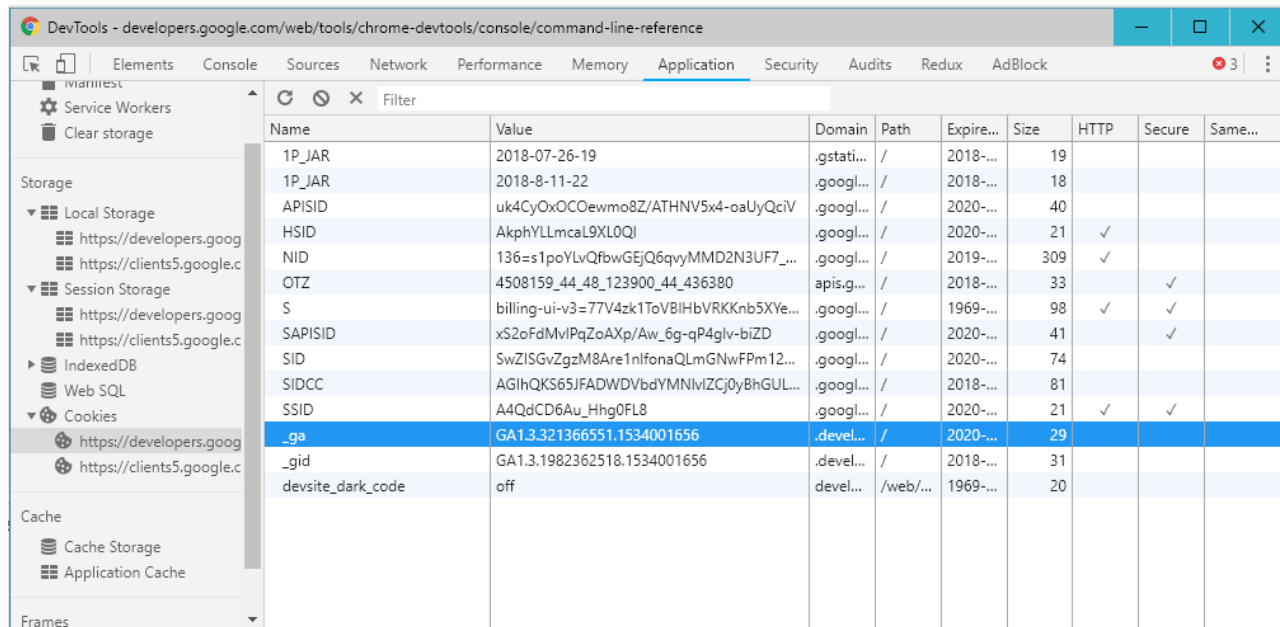


"The most effective debugging tool is still careful thought, coupled with judiciously placed print statements."

Sources panel and debugging running code



Application data

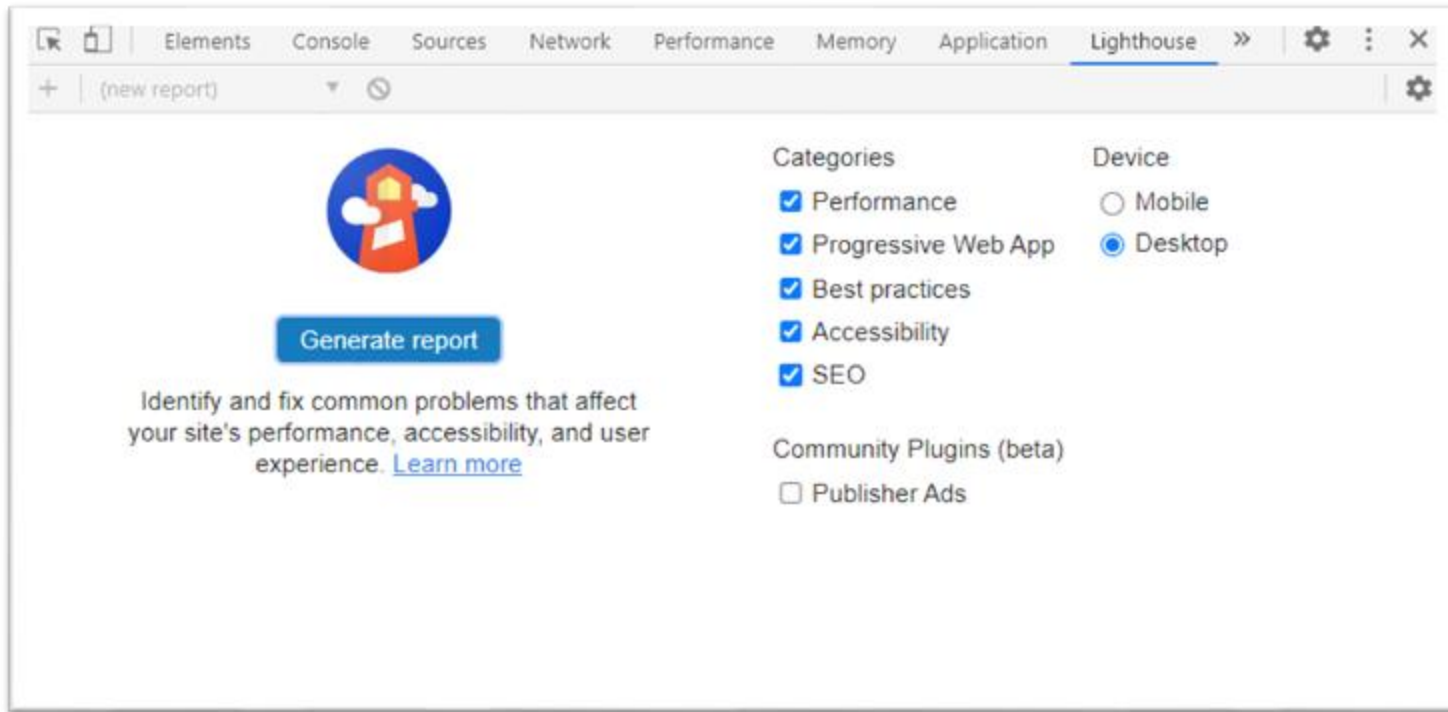


The screenshot shows the Chrome DevTools Application tab. The left sidebar displays the 'Storage' section, which is expanded to show 'Local Storage', 'Session Storage', 'IndexedDB', 'Web SQL', 'Cookies', and 'Cache'. The 'Cookies' section is selected, and the main pane shows a table of cookies. The table has columns: Name, Value, Domain, Path, Expire..., Size, HTTP, Secure, and Same... The cookie named '_ga' is highlighted in blue.

| Name | Value | Domain | Path | Expire... | Size | HTTP | Secure | Same... |
|-------------------|---|------------|----------|-----------|------|------|--------|---------|
| 1P_JAR | 2018-07-26-19 | .gstati... | / | 2018-... | 19 | | | |
| 1P_JAR | 2018-8-11-22 | .googl... | / | 2018-... | 18 | | | |
| APISID | uk4CyOxOCOewmo8Z/ATHNV5x4-0aUyQciV | .googl... | / | 2020-... | 40 | | | |
| HSID | AkphYLLmcaL9XL0QI | .googl... | / | 2020-... | 21 | ✓ | | |
| NID | 136=s1poYlvQfbwGEjQ6qvyMMD2N3UF7_... | .googl... | / | 2019-... | 309 | ✓ | | |
| OTZ | 4508159_44_48_123900_44_436380 | apis.g... | / | 2018-... | 33 | | ✓ | |
| S | billing-ui-v3=77V4zk1ToVBtHbVRKKnb5XYe... | .googl... | / | 1969-... | 98 | ✓ | ✓ | |
| SAPISID | xS2oFdMvIPqZoAXp/Aw_6g-qP4glv-biZD | .googl... | / | 2020-... | 41 | | ✓ | |
| SID | SwZISGvZgzM8Are1nlfonaQLmGNwFPm12... | .googl... | / | 2020-... | 74 | | | |
| SIDCC | AGlhQKS65JFADWDVbdYMNlvZCj0yBhGUL... | .googl... | / | 2018-... | 81 | | | |
| SSID | A4QdCD6Au_Hhg0FL8 | .googl... | / | 2020-... | 21 | ✓ | ✓ | |
| _ga | GA1.3.321366551.1534001656 | .devel... | / | 2020-... | 29 | | | |
| _gid | GA1.3.1982362518.1534001656 | .devel... | / | 2018-... | 31 | | | |
| devsite_dark_code | off | devel... | /web/... | 1969-... | 20 | | | |

- HTML 5 Database
- Local Storage
- Cookies
- AppCache

Lighthouse



Console

The screenshot displays a web browser's developer console with the following components:

- Elements Panel:** Shows the DOM tree with the following structure:

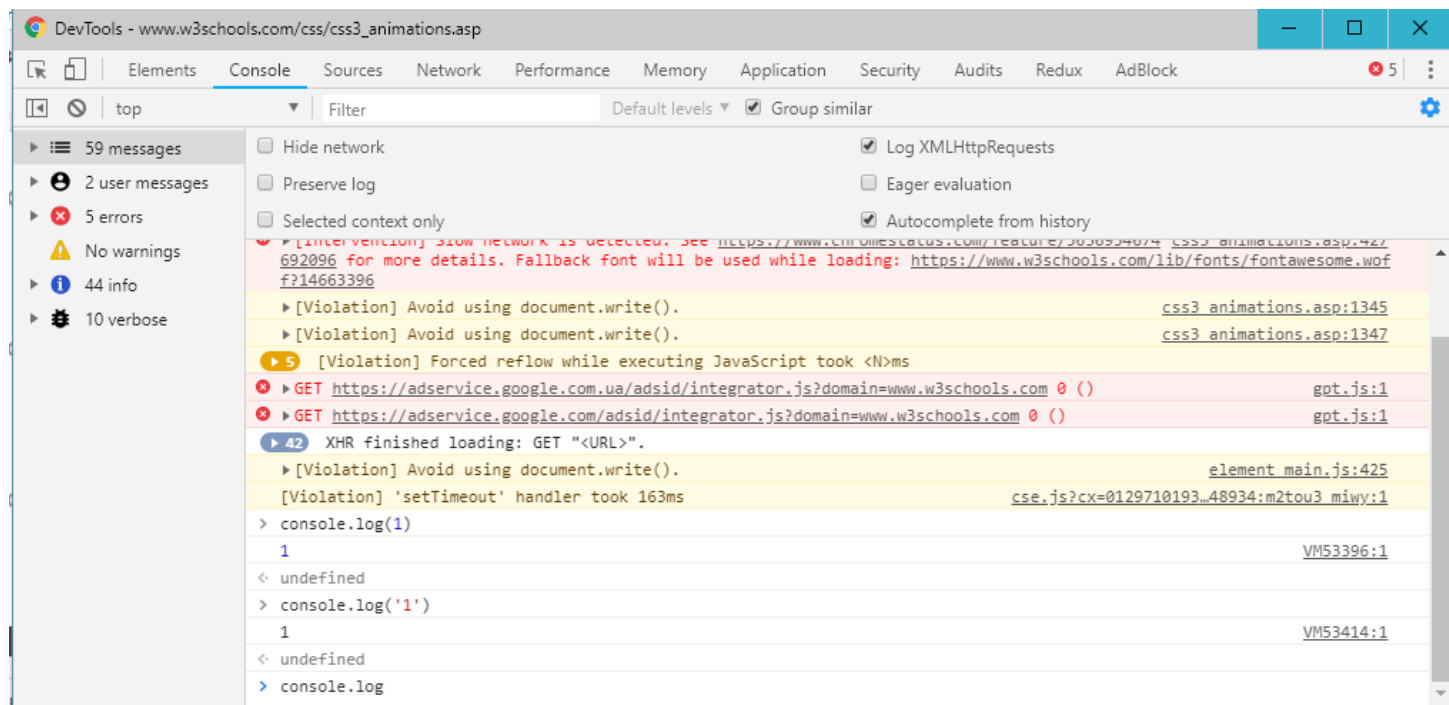
```
contextMenu=edit-history-menu data-search-uri class="document">
  <script>...</script>
  <ul id="nav-access">...</ul>
  <!-- Header -->
  <header id="main-header" class="header-main">...</header>
  <!-- Content will go here -->
  <main id="content" role="main">
    <!-- heading -->
    <div id="wiki-document-head" class="document-head">...
  </div>
  <div class="center clear">
    <div class="wiki-main-content" id="document-main">
      <div class="center">
```
- Styles Panel:** Shows the CSS rules for the selected element:

```
element.style {
}

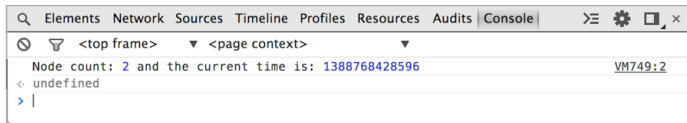
@media (min-width: 47.9385em)
.column-container.column-container-reverse[class^=column-]:first-child {
  margin-right: 0;
}

@media (min-width: 47.9385em)
.column-container.column-container-reverse[class^=column-] {
  float: right;
```
- Console Panel:** Shows two warnings:
 - Warning 1: The resource <https://developer.mozilla.org/static/fonts/locales/ZillaSlab-Regular.subset.bbc33fb47cf6.woff2> was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.
 - Warning 2: The resource <https://developer.mozilla.org/static/fonts/locales/ZillaSlab-Bold.subset.e96c15f68c68.woff2> was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

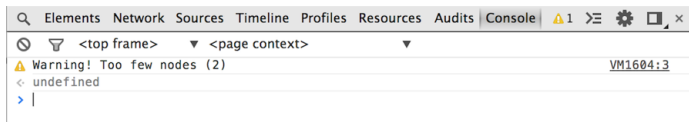
Console - Settings



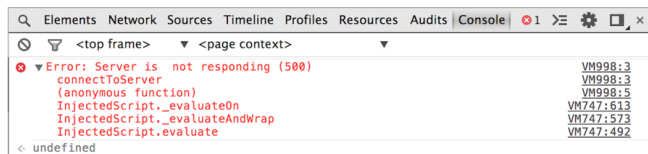
Console - API



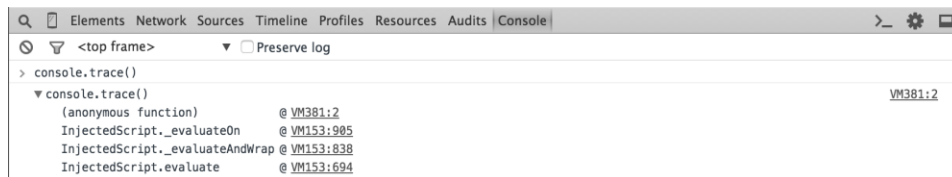
`console.log()`



`console.warn()`



`console.assert()`
`console.error()`

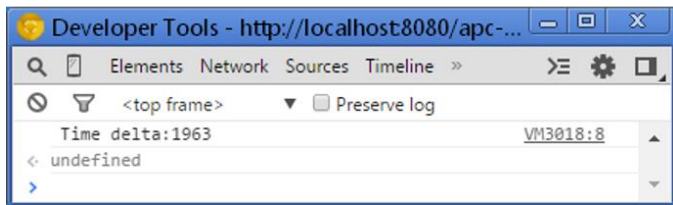


`console.trace()`

Console – Measuring running time

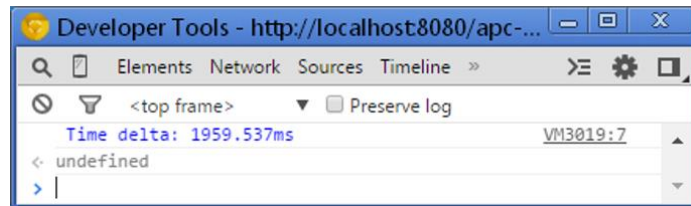
Old school

```
let t = new Date();
let array = new Array(1000000);
for (let i = array.length - 1; i >= 0; i--) {
    array[i] = {}
}
t = new Date() - t;
console.log('Time delta:' + t);
```

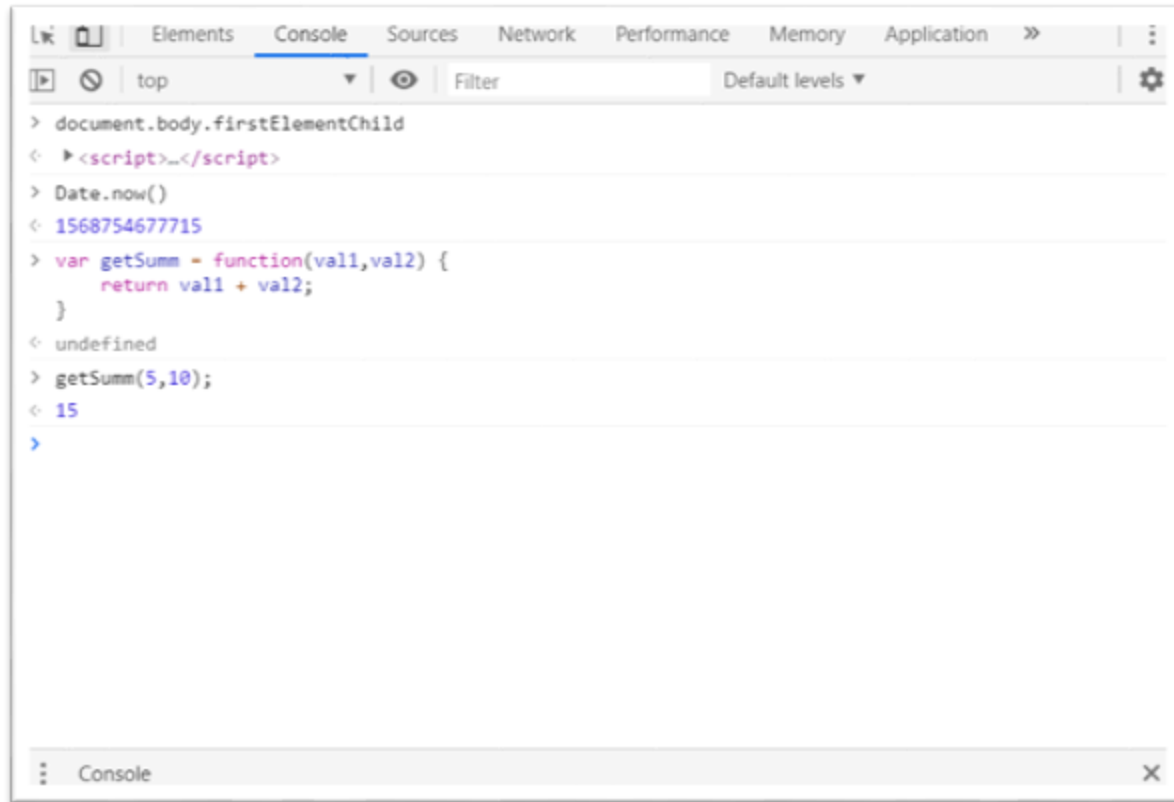


Console API

```
console.time('Time delta');
let array = new Array(1000000);
for (let i = array.length - 1; i >= 0; i--) {
    array[i] = {}
}
t = new Date() - t;
console.timeEnd('Time delta');
```



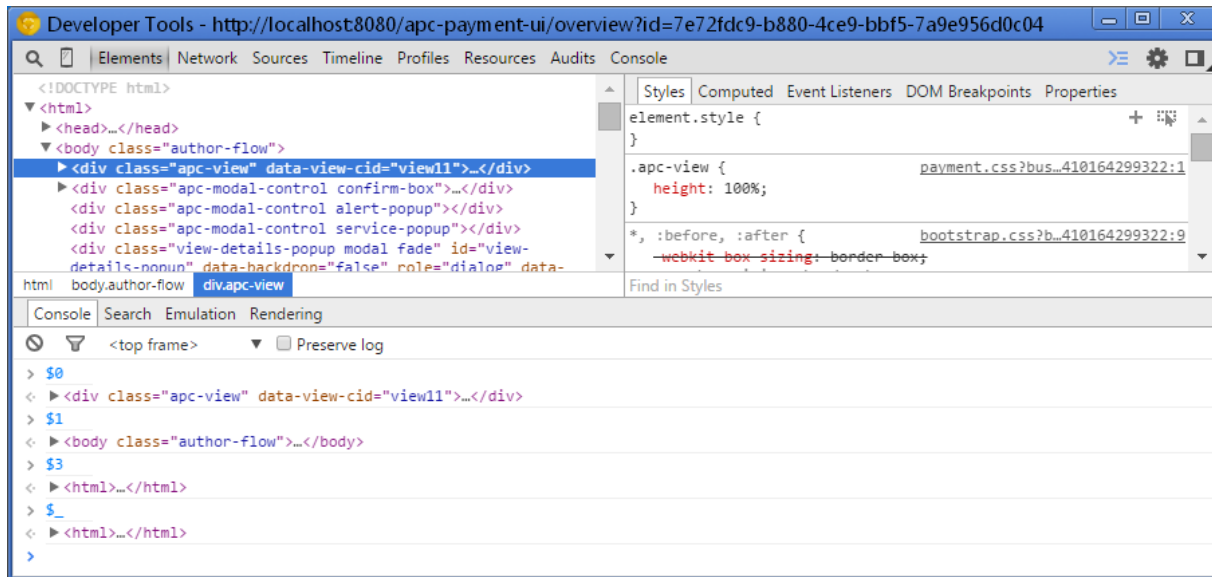
Console - Evaluating expressions



Console – Selecting elements

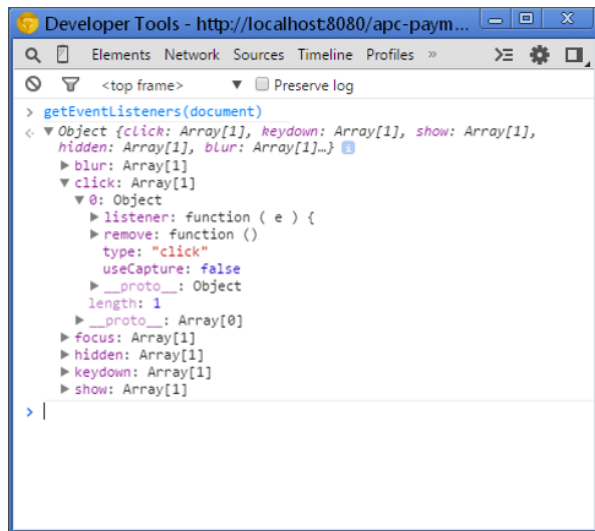
```
$0, $1, $2, $3, $4  
$_
```

```
$('.article')  
$$('.article')
```

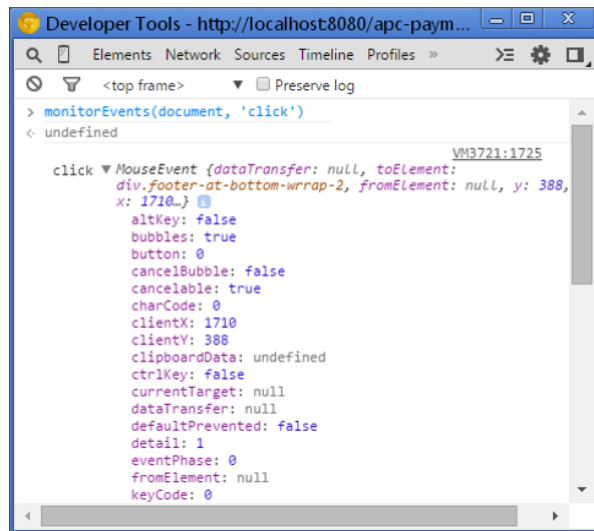


Console - Monitoring events

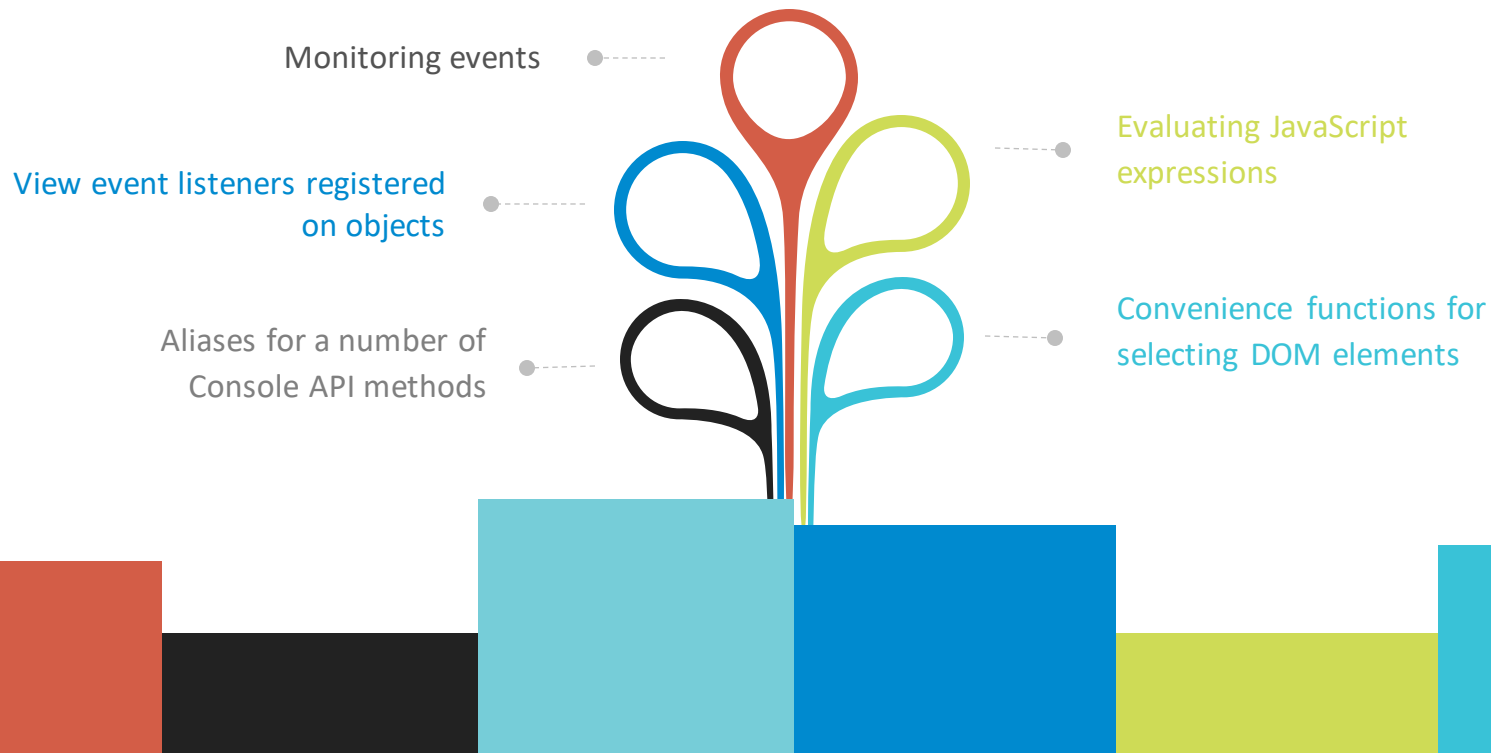
`getEventListeners(object)`



`monitorEvents(window, "resize")`
`unmonitorEvents(object[, events])`

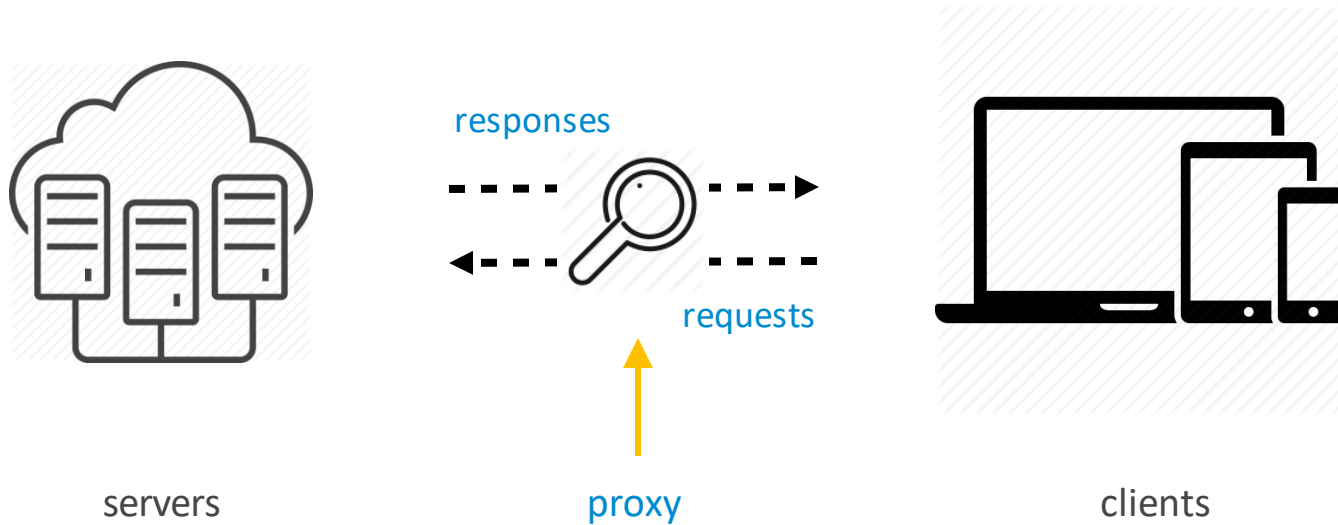


Console - Utilities



FIDDLER / CHARLES PROXY

Fiddler, Charles Proxy



Fiddler – AutoResponder

The screenshot displays the Fiddler Everywhere application interface. The top bar includes 'View', 'Help', and the 'Fiddler Everywhere' logo with an 'Upgrade' button. The user profile 'Stanislav Dolgachov' is visible in the top right.

The main interface is divided into several sections:

- Sessions:** A sidebar on the left showing 'My Sessions' and 'Shared with Me'.
- Live Traffic (Capturing):** A central panel showing a list of captured requests. The 'Composer' tab is active, displaying a table of requests.
- AutoResponder:** A panel on the right with a toggle switch set to 'On'. It contains a table for defining rules.

Live Traffic Table:

#	URL	Result	Method
1	http://euc-powerpoint.officeapps.live.C...	200	CONNECT
2	http://c1-powerpoint-15.cdn.office.net...	200	CONNECT
3	http://d27xe7juh1us6.cloudfront.net...	200	CONNECT
4	http://westus2-prod-2.notifications.tea...	200	CONNECT
5	http://westeurope.notifications.teams....	200	CONNECT
6	http://browser.pipe.aria.microsoft.co...	200	CONNECT
7	http://azscus1-client-s.gateway.messe...	200	CONNECT
8	http://outlook.office.com:443	200	CONNECT
9	http://accounts.google.com:443	200	CONNECT
10	http://www.google.com:443	200	CONNECT
11	http://www.google.com:443	200	CONNECT
12	http://localhost:17444/	502	GET
13	http://encrypted-tbn0.gstatic.com:443	200	CONNECT
14	http://localhost:17444/	502	GET
15	http://localhost:17444/favicon.ico	502	GET

AutoResponder Table:

Match Condition	Action
localhost:17444	http://www.example.com

The bottom status bar shows 'My Resources' and a 'Connected' indicator.

FIDDLER – LIST OF FEATURES

1

See all requests, headers, cookies and parameters transferred to / from the server.

2

Test the site on slow connection with the Internet

3

Rewrite transferred data / headers

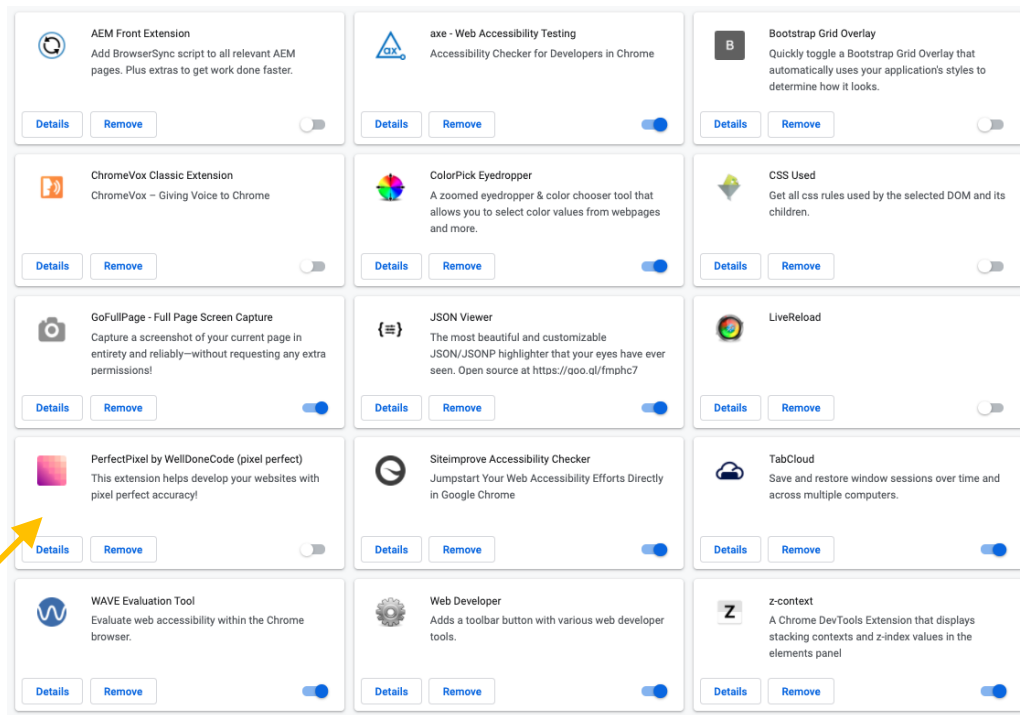
NIGHTLY BROWSERS, EXTENSIONS

Chrome extensions

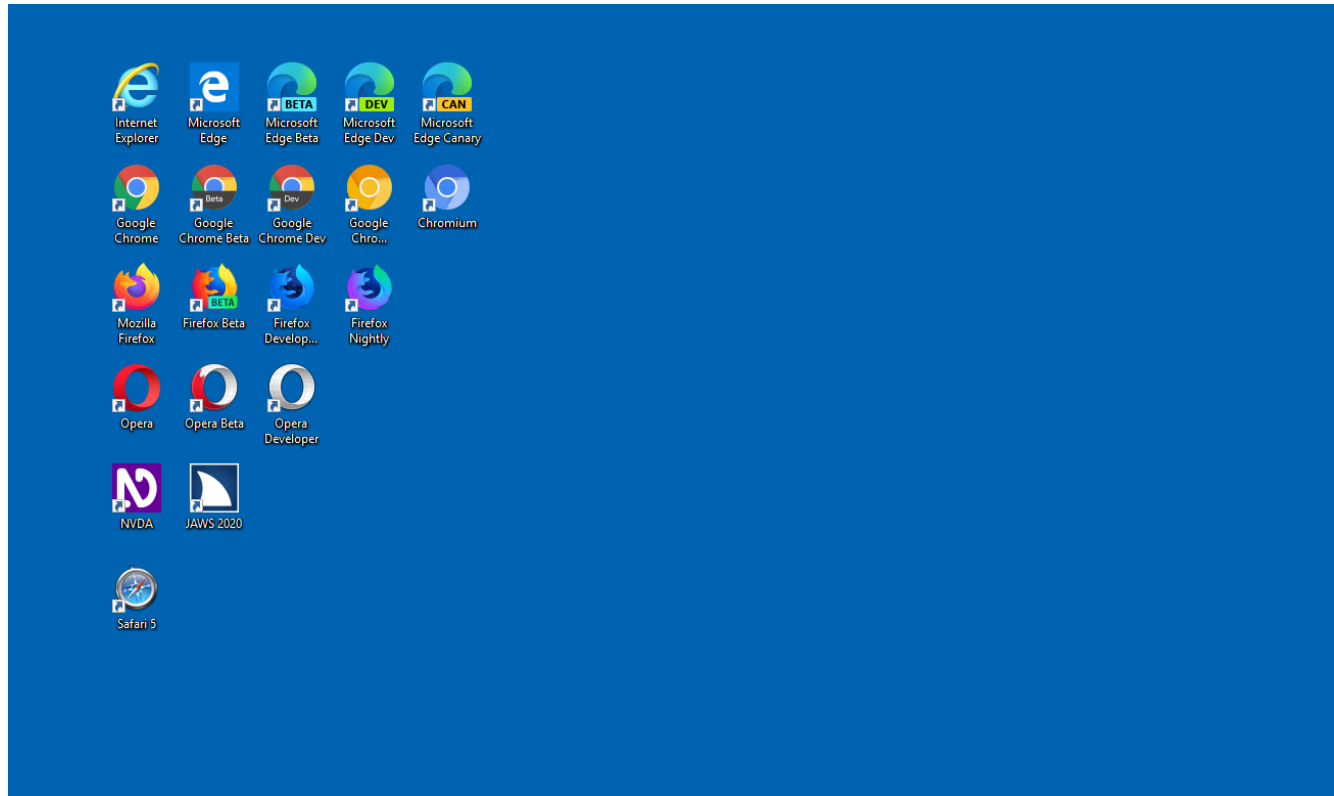
Always align extensions with
your **project policies**!

Sometimes you simply cannot
install extensions, sometimes
it is possible, however, these
can pose a security risk.

That one is really cool!



Nightly Browsers and Screen Readers





That's all Folks!

Debugging tools – wrap up

Easy To Use

Intuitively clear
interface



See the errors

We can see errors during
execution our code

Performance

Opportunity to check
performance



State

Allows to see the state
at the time of debugging

Accuracy

Pinpoints places and exact
description of a problem



Monitoring

We can trace events

Q&A