JavaScript Debugging

Sergio Tabares Hernández

Adrián Epifanio Rodríguez Hernández

Index

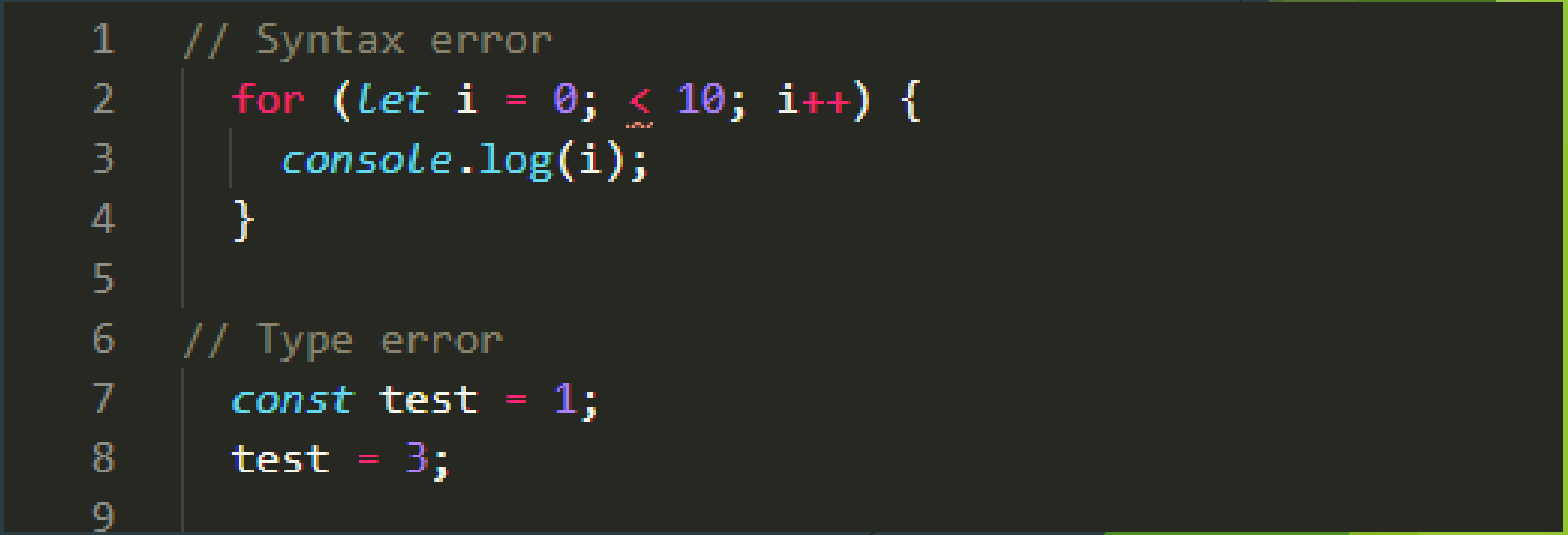
1. What is a debugger?

- Program used to test and run the target program under controlled conditions that permit the programmer monitoring the changes while it runs that may indicate malfunctioning code.

Visual Studio Code

1.1 Syntax or type errors

- These are always caught by the compiler, and reported via error messages. Typically, an error message clearly indicates the cause of error; for instance, the line number, the incorrect piece of code, and an explanation.



```
// Syntax error
for (let i = 0; < 10; i++) {
  console.log(i);
}

// Type error
const test = 1;
test = 3;

```

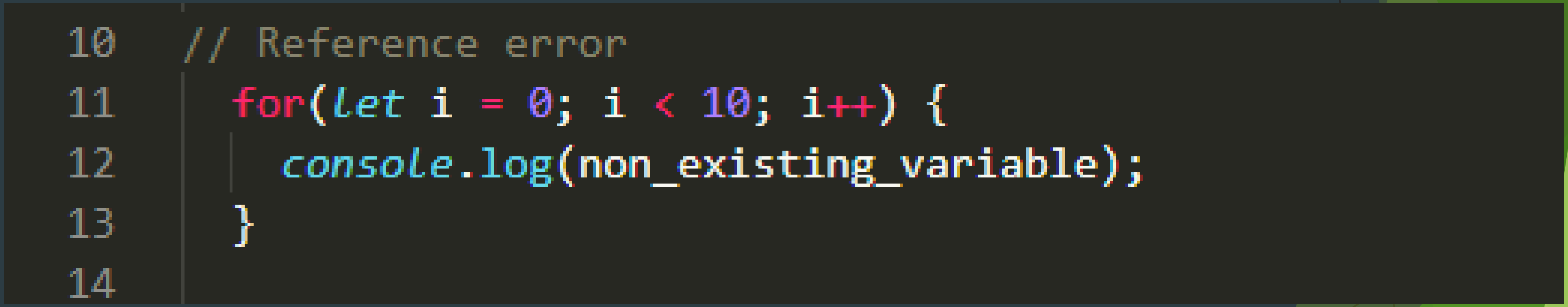
1.2 Typos and other simple errors

- That have pass undetected by the type-checker or the other checks in the compiler. Once these are identified, they can easily be fixed. Passing parameters in incorrect order, or using the wrong element order in tuples.

  `x + y * z` instead of `(x + y) * z;`

1.3 Reference errors

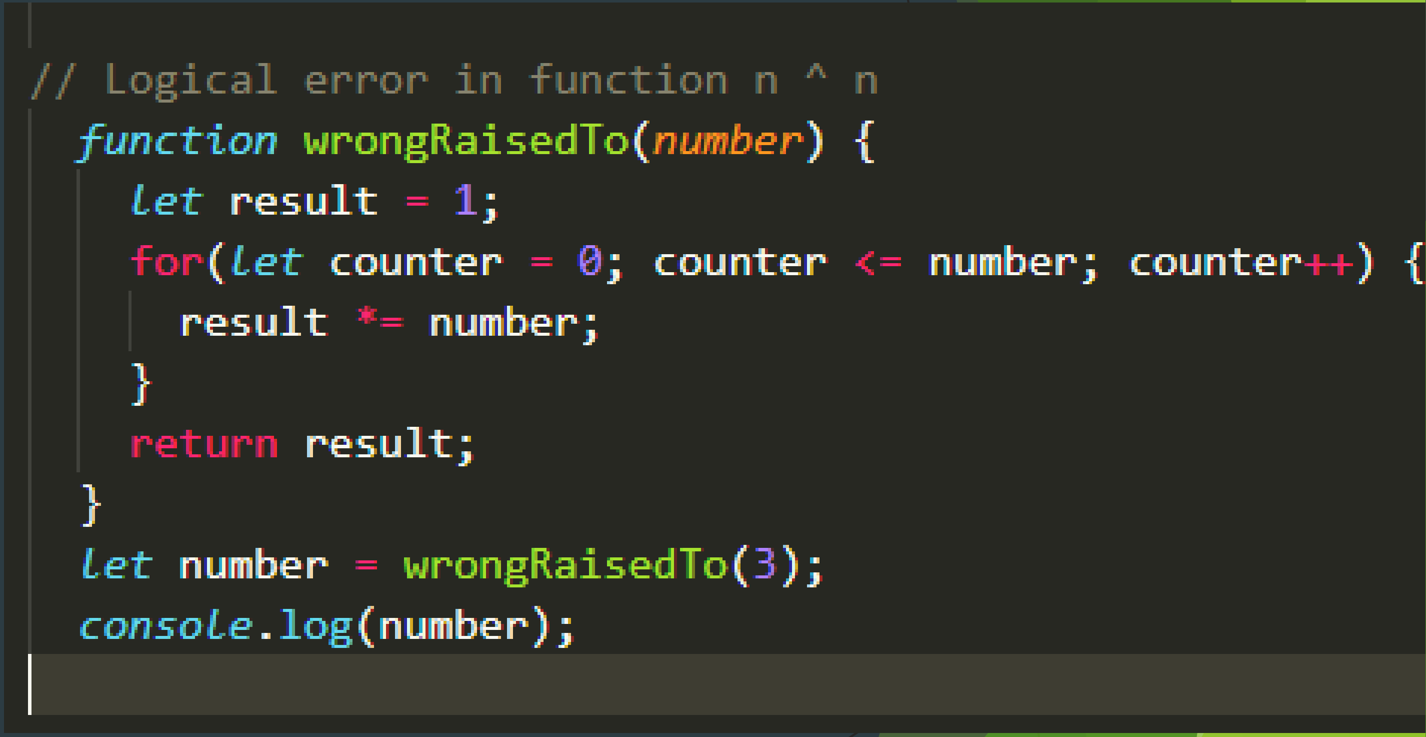
- Represents an **error** when a non-existent variable is referenced.



```
// Reference error
for(let i = 0; i < 10; i++) {
  console.log(non_existing_variable);
}

```

1.4 Implementarion and logical errors

- It may be the case that logic in the high-level algorithm of a program is correct, but some low-level, concrete data structures are being manipulated incorrectly, breaking some internal representation invariants. If the algorithm is logically flawed, the programmer must re-think the algorithm. Fixing such problems is more difficult, especially if the program fails on just a few corner cases.



```
// Logical error in function n ^ n
function wrongRaisedTo(number) {
  let result = 1;
  for(let counter = 0; counter <= number; counter++) {
    result *= number;
  }
  return result;
}
let number = wrongRaisedTo(3);
console.log(number);
```

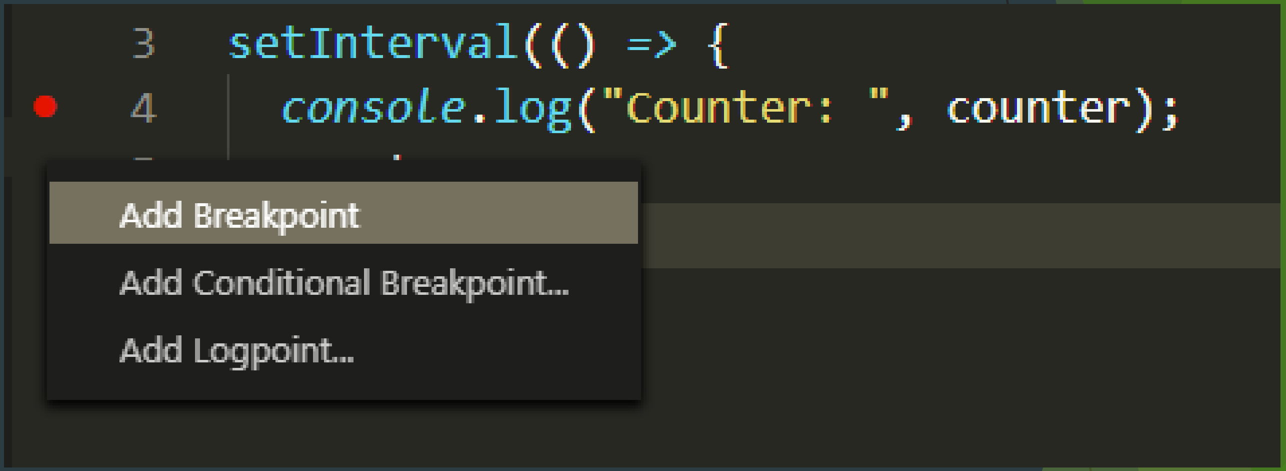
2. Strategies

1. Incremental and bottom-up program development
2. Instrument program with assertions
3. Use debuggers
4. Backtracking
5. Binary search
6. Problem simplification
7. Scientific method: form hypotheses
8. Bug clustering

3. Breakpoints / Logpoints

- A line where the program prints a message or stops till the user clicks to continue the execution.

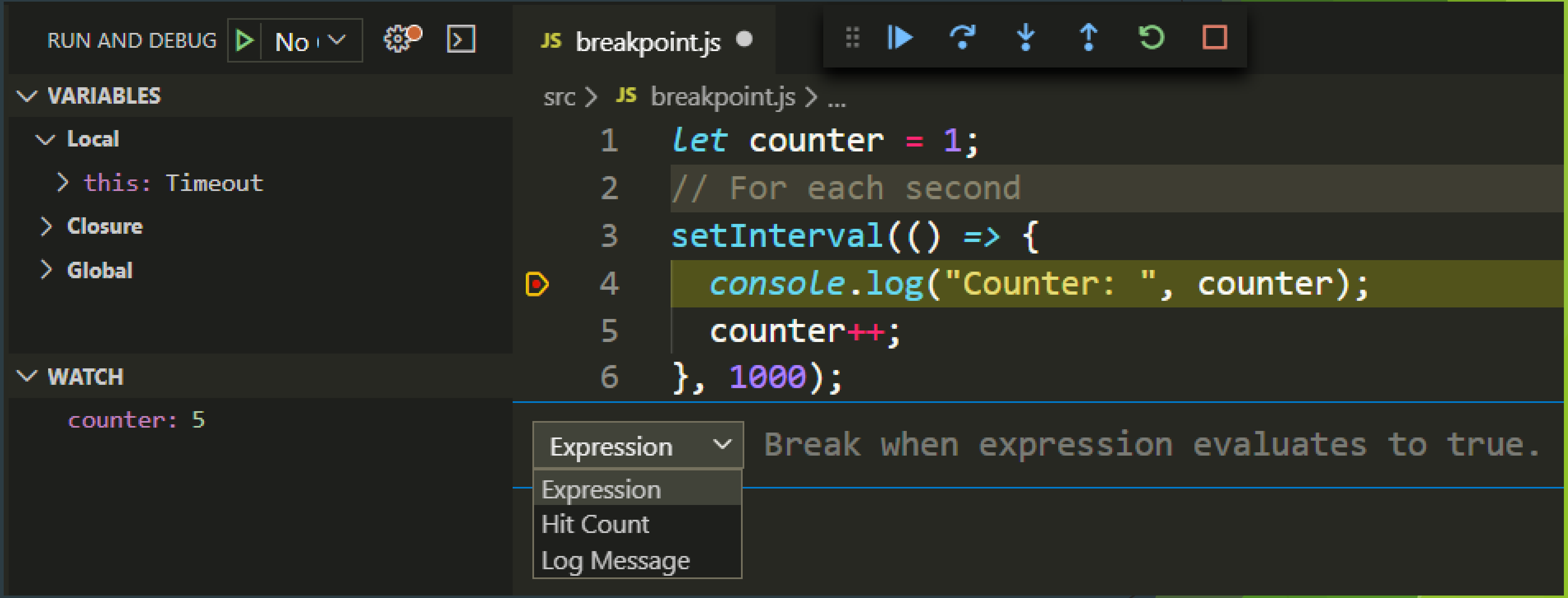
- They are used for controlling the variables while the program is running helping the programmer to find errors in the code.



```
setInterval(() => {
  console.log("Counter: ", counter);
```

3.1 Breakpoints / Logpoints Types

- Normal
- Conditional
- Hit count
- Log message

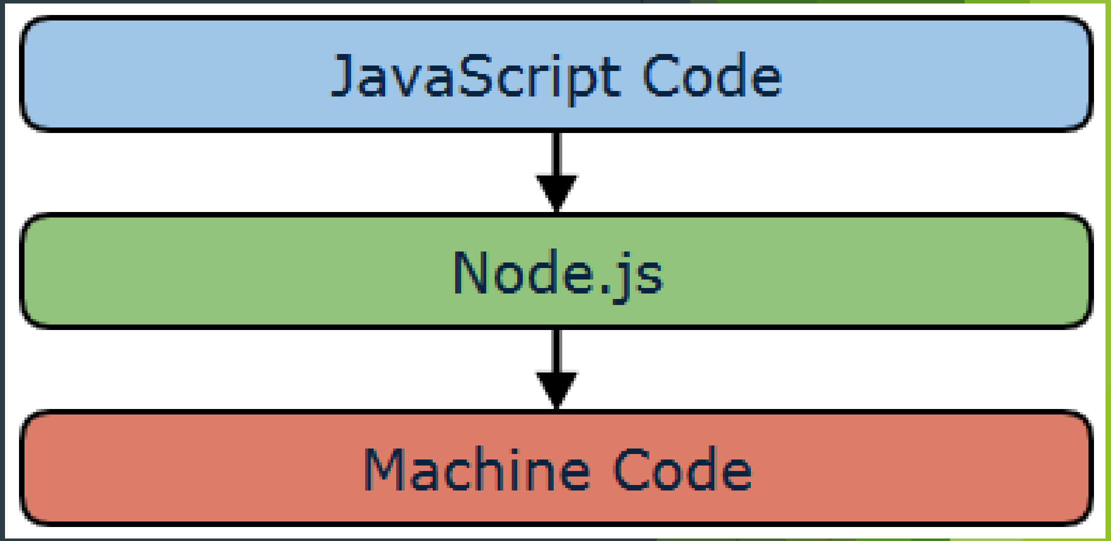


```
this: Timeout
```

```
counter: 5
```

```javascript
let counter = 1;
// For each second
setInterval(() => {
  console.log("Counter: ", counter);
  counter++;
}, 1000);
```

4. Node.js

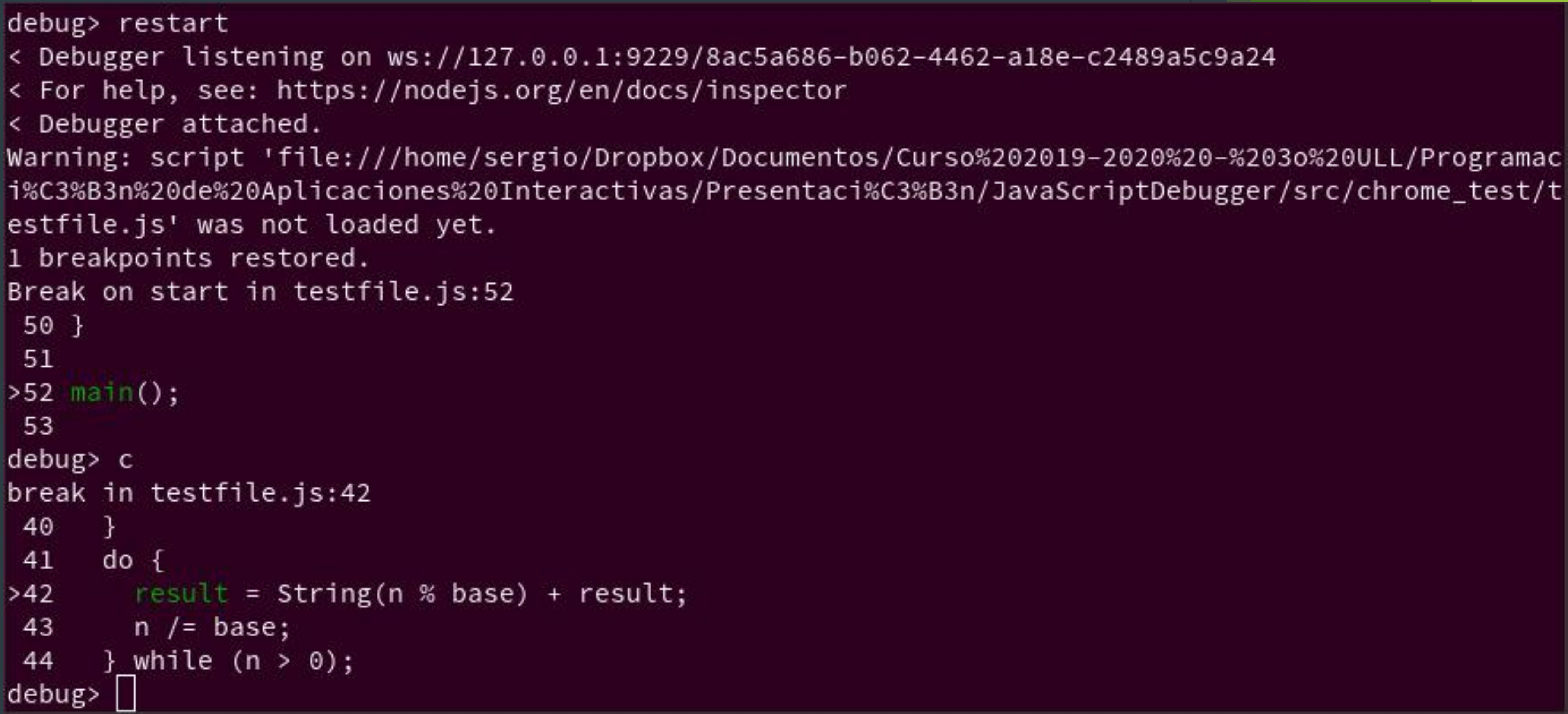


4.1 Stepping

- cont, c: Continue execution
- next, n: Step next
- step, s: Step in
- out, o: Step out
- pause: Pause running code

4.2 Execution control

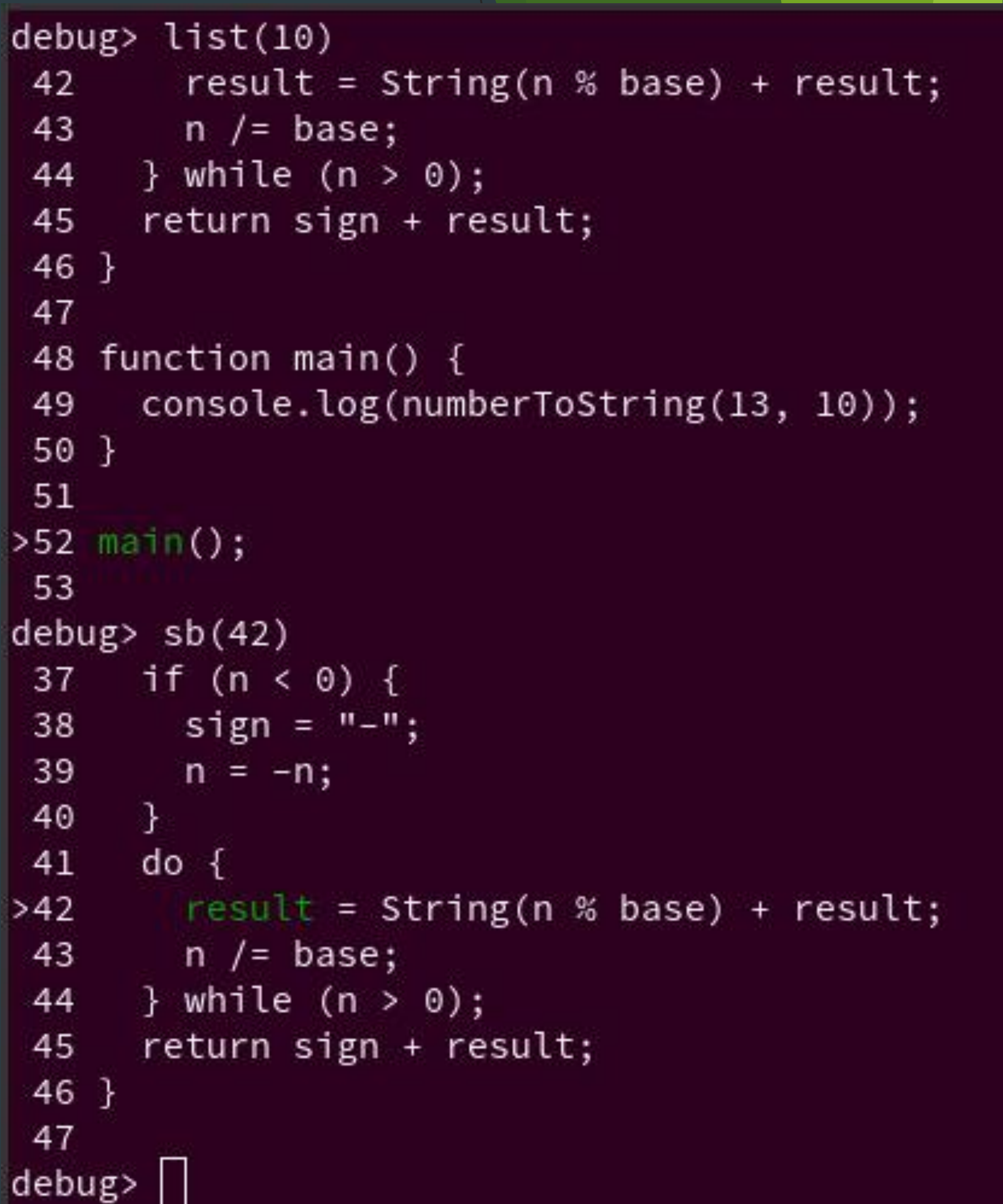
- Run
- Restart
- Kill



```
debug> restart
< Debugger listening on ws://127.0.0.1:9229/8ac5a686-b062-4462-a18e-c2489a5c9a24
< For help, see: https://nodejs.org/en/docs/inspector
< Debugger attached.
Warning: script 'file:///home/sergio/Dropbox/Documentos/Curso%202019-2020%20-%203o%20ULL/Programac
i%C3%B3n%20de%20Aplicaciones%20Interactivas/Presentaci%C3%B3n/JavaScriptDebugger/src/chrome_test/t
estfile.js' was not loaded yet.
1 breakpoints restored.
Break on start in testfile.js:52
 50 }
 51
>52 main();
 53
debug> c
break in testfile.js:42
 40   }
 41   do {
>42     result = String(n % base) + result;
 43     n /= base;
 44   } while (n > 0);
debug> 
```

4.3 Breakpoints

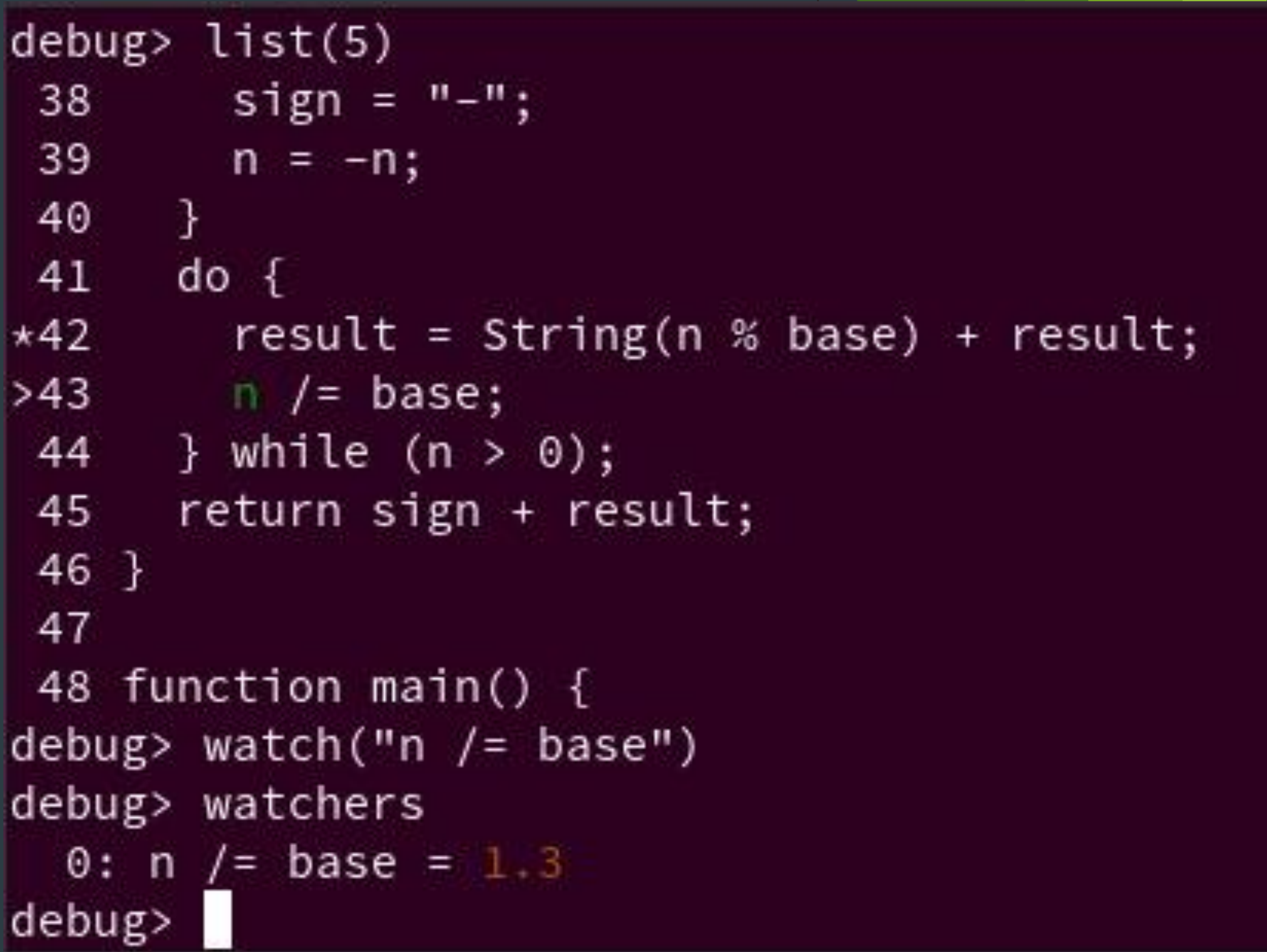
- setBreakpoint(), sb()
- setBreakpoint(line), sb(line)
- setBreakpoint('fn()'), sb(...)
- setBreakpoint('script.js', 1), sb(...)
- clearBreakpoint('script.js', 1), cb(...)



```
debug> list(10)
 42       result = String(n % base) + result;
 43       n /= base;
 44     } while (n > 0);
 45     return sign + result;
 46 }
 47
 48 function main() {
 49   console.log(numberToString(13, 10));
 50 }
 51
>52 main();
 53
debug> sb(42)
 37   if (n < 0) {
 38     sign = "-";
 39     n = -n;
 40   }
 41   do {
>42     result = String(n % base) + result;
 43     n /= base;
 44   } while (n > 0);
 45   return sign + result;
 46 }
 47
debug> 
```

4.4 Information

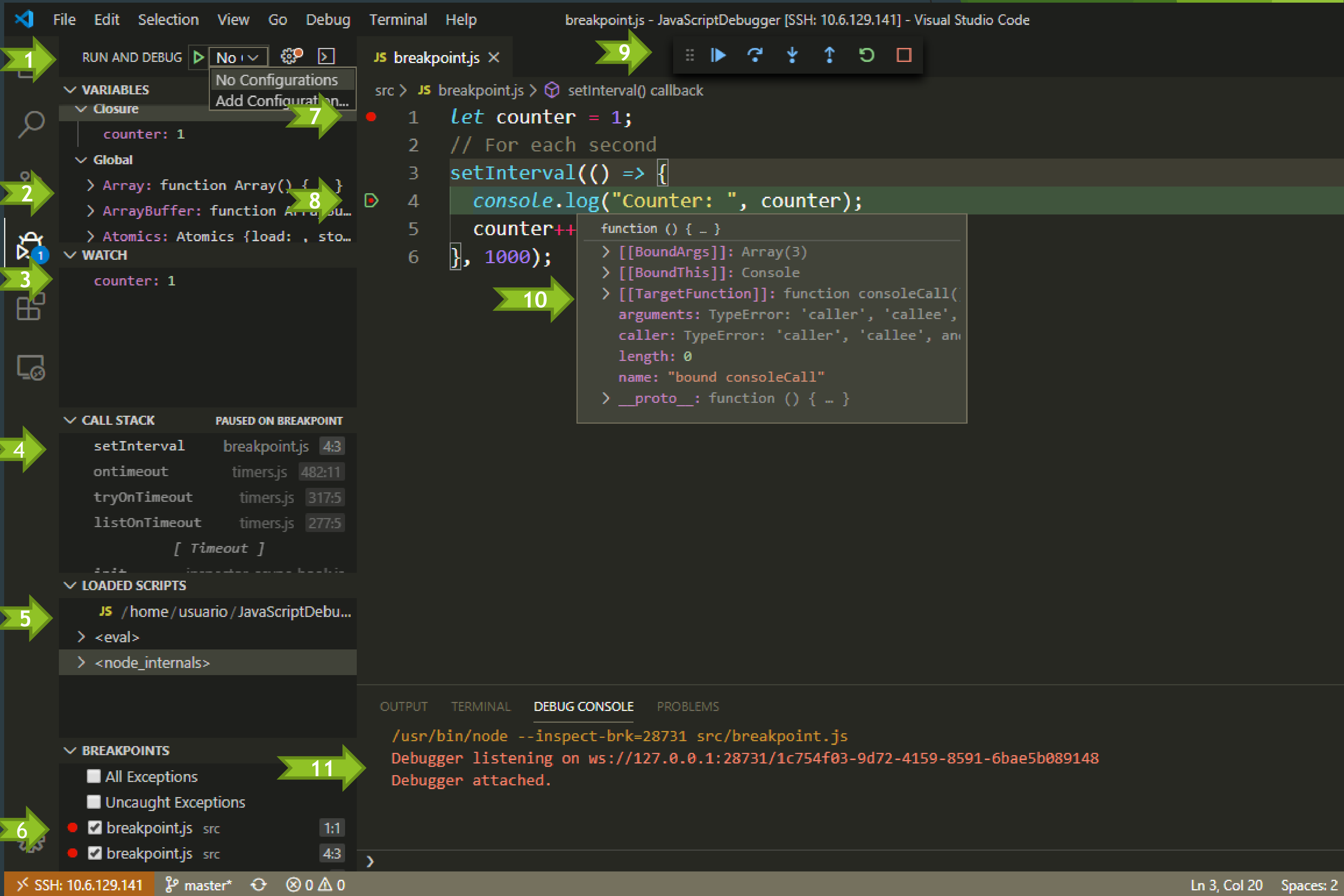
- backtrace, bt
- list(5)
- watch(expr)
- unwatch(expr)
- watchers
- repl
- exec expr



```
debug> list(5)
 38       sign = "-";
 39       n = -n;
 40     }
 41     do {
*42       result = String(n % base) + result;
>43       n /= base;
 44     } while (n > 0);
 45     return sign + result;
 46 }
 47
 48 function main() {
debug> watch("n /= base")
debug> watchers
  0: n /= base = 1.3
debug> 
```

5. Visual Studio Code

1. Debug selector
2. Navigator
3. Watch expresión
4. Stack traces
5. Debugging scripts
6. Breakpoints list
7. Breakpoint line
8. Program pointer
9. Debug Actions
10. Variable values
11. Debug console



```
counter: 1
> Array: function Array() { … }
> ArrayBuffer: function ArrayBu…
> Atomics: Atomics {load: , sto…
```

```
counter: 1
```

setInterval	breakpoint.js	4:3
ontimeout	timers.js	482:11
tryOnTimeout	timers.js	317:5
listOnTimeout	timers.js	277:5
[Timeout]		

JS /home/usuario/JavaScriptDebu…
> <eval>
> <node_internals>

● ☑	breakpoint.js	src	1:1
● ☑	breakpoint.js	src	4:3

src > JS breakpoint.js > setInterval() callback

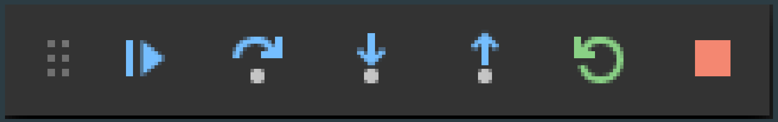
```javascript
let counter = 1;
// For each second
setInterval(() => {
  console.log("Counter: ", counter);
  counter++
}, 1000);
```

```
function () { … }
> [[BoundArgs]]: Array(3)
> [[BoundThis]]: Console
> [[TargetFunction]]: function consoleCall(
  arguments: TypeError: 'caller', 'callee',
  caller: TypeError: 'caller', 'callee', an
  length: 0
  name: "bound consoleCall"
> __proto__: function () { … }
```

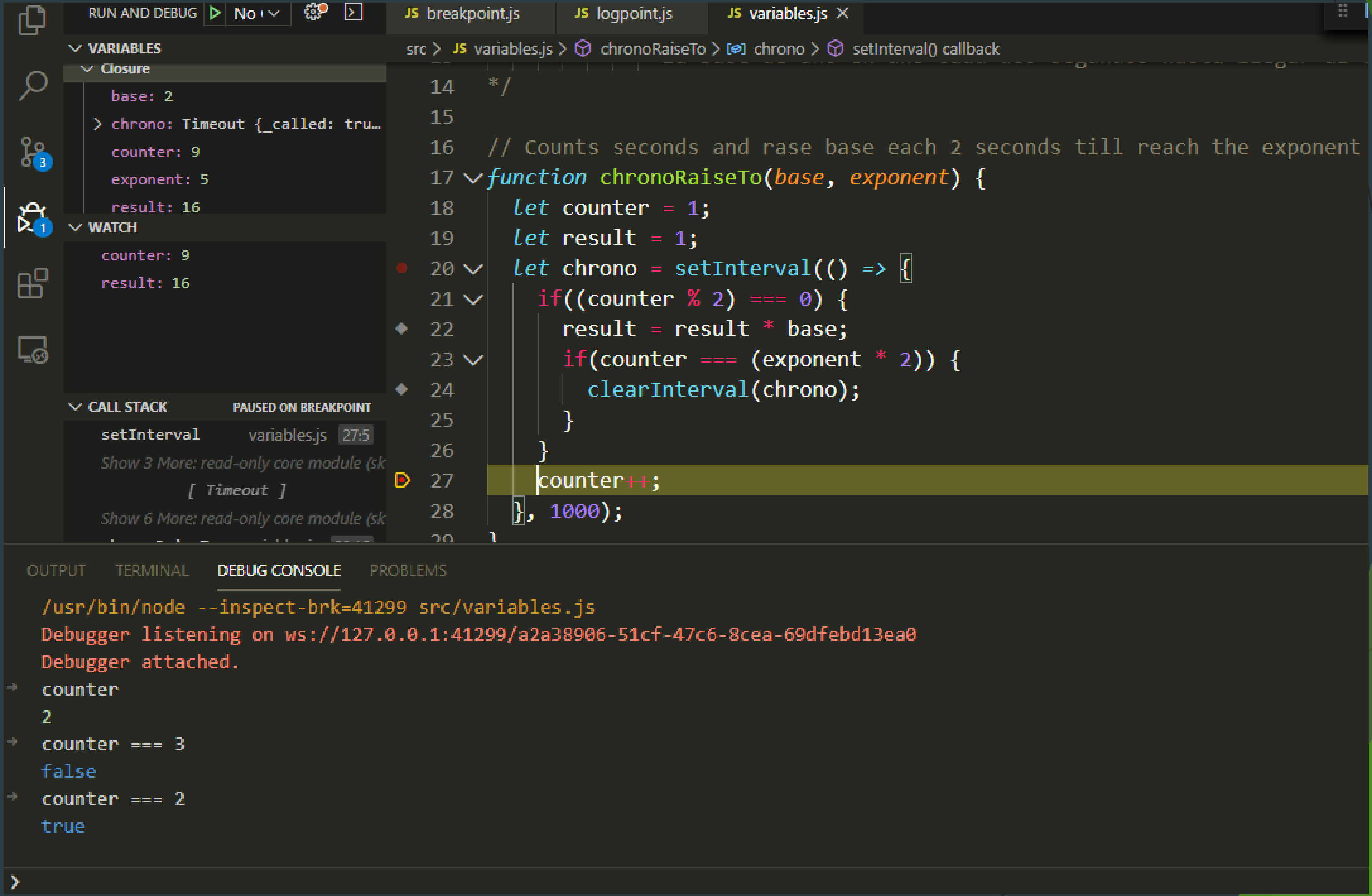
```
/usr/bin/node --inspect-brk=28731 src/breakpoint.js
Debugger listening on ws://127.0.0.1:28731/1c754f03-9d72-4159-8591-6bae5b089148
Debugger attached.
```

5.1 Keyboard shortcuts

Continue / Pause	F5
Step Over	F10
Step Into	F11
Step Out	Shift + F11
Restart	Ctrl + Shift + F5
Stop	Shift + F5
Add Breakpoint	F9



5.2 Variables



```
base: 2
> chrono: Timeout {_called: tru…
counter: 9
exponent: 5
result: 16
```

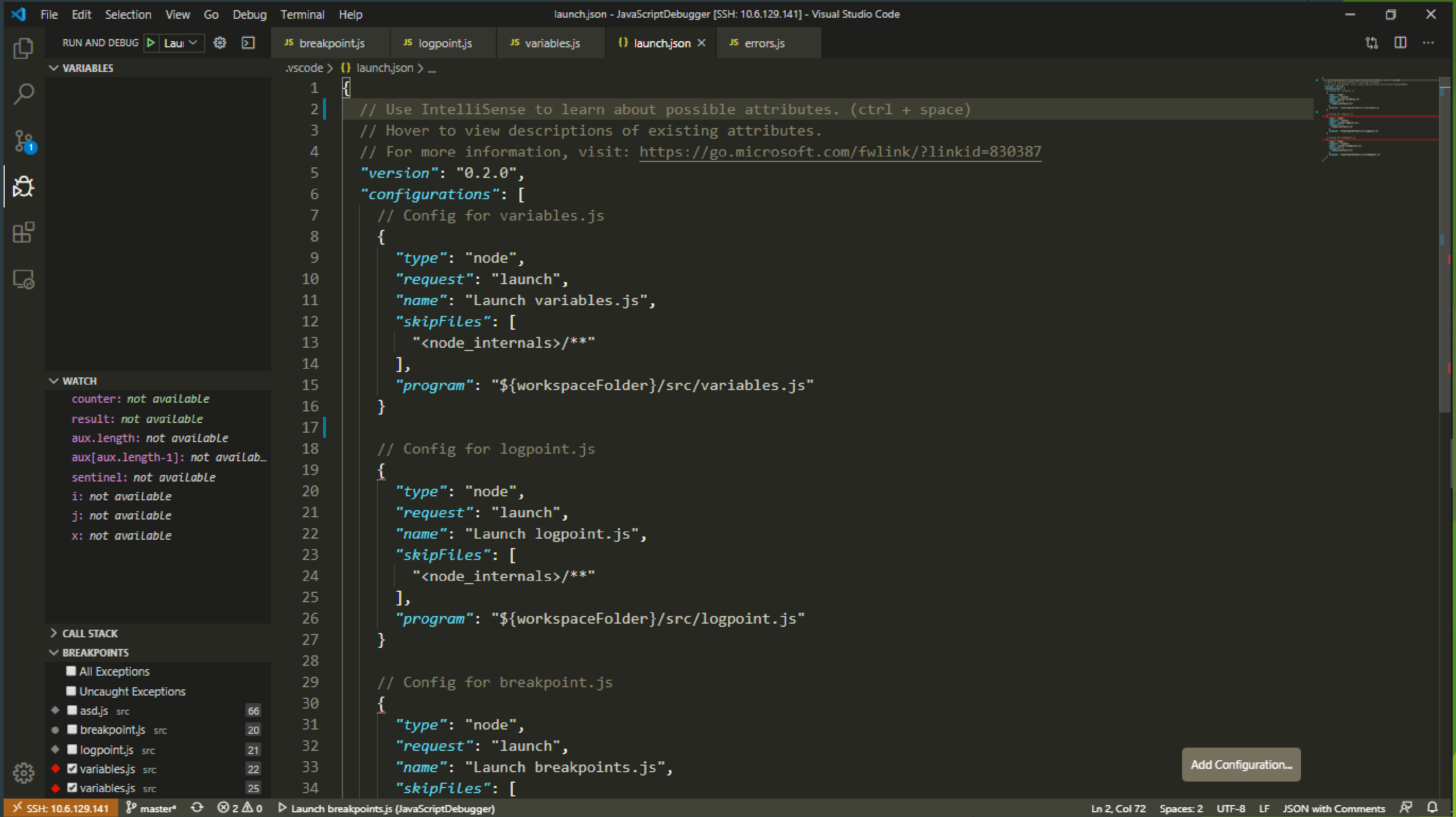
```
counter: 9
result: 16
```

```
setInterval        variables.js 27:5
Show 3 More: read-only core module (sk
[ Timeout ]
Show 6 More: read-only core module (sk
```

```javascript
*/

// Counts seconds and rase base each 2 seconds till reach the exponent
function chronoRaiseTo(base, exponent) {
  let counter = 1;
  let result = 1;
  let chrono = setInterval(() => {
    if((counter % 2) === 0) {
      result = result * base;
      if(counter === (exponent * 2)) {
        clearInterval(chrono);
      }
    }
    counter++;
  }, 1000);
```

```
/usr/bin/node --inspect-brk=41299 src/variables.js
Debugger listening on ws://127.0.0.1:41299/a2a38906-51cf-47c6-8cea-69dfebd13ea0
Debugger attached.
counter
2
counter === 3
false
counter === 2
true
```

5.3 Launch.json

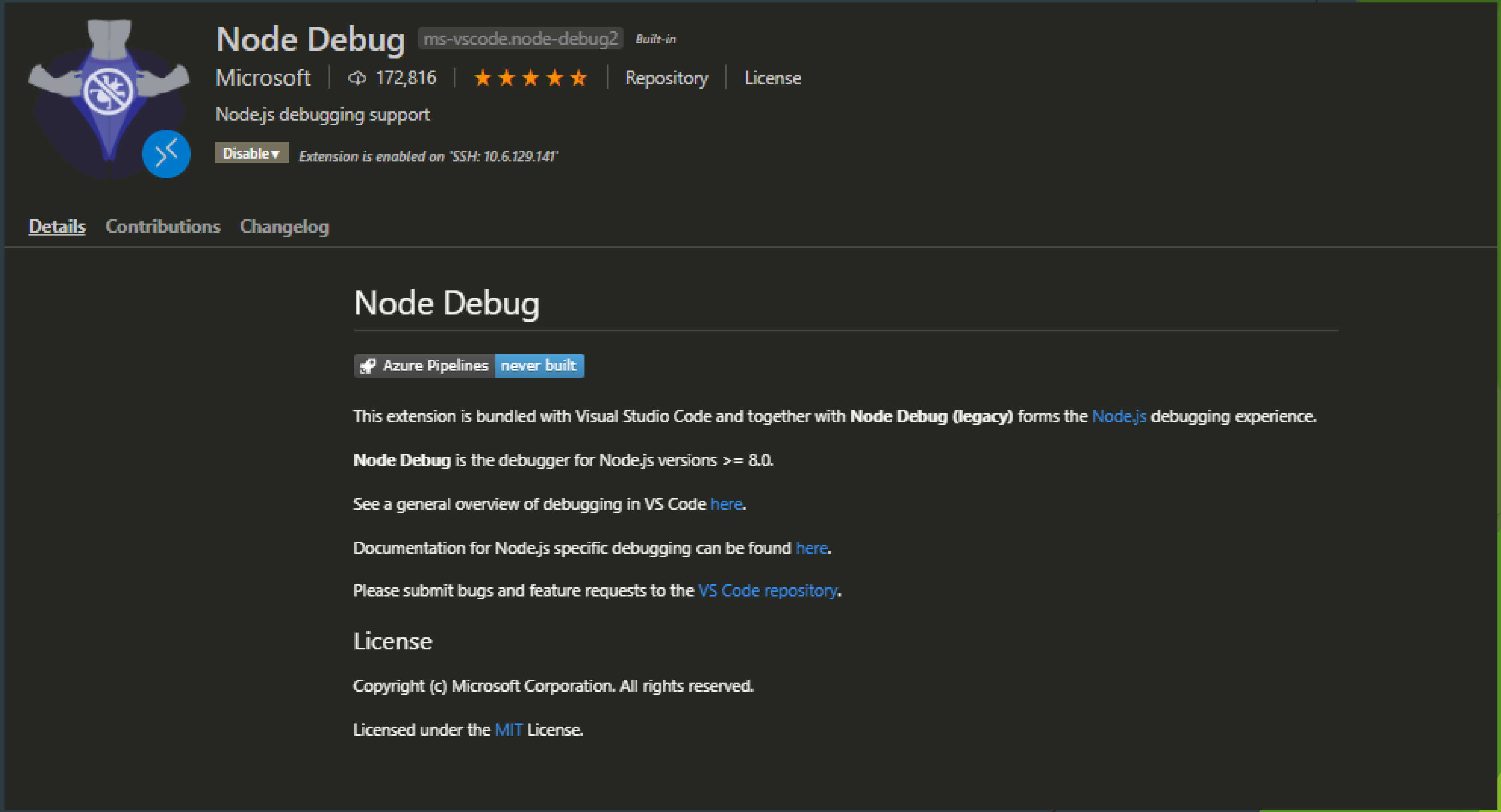

File Edit Selection View Go Debug Terminal Help
launch.json - JavaScriptDebugger [SSH: 10.6.129.141] - Visual Studio Code
RUN AND DEBUG
breakpoint.js
logpoint.js
variables.js
launch.json
errors.js
.vscode > launch.json > ...
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Add Configuration...
SSH: 10.6.129.141
master*
Launch breakpoints.js (JavaScriptDebugger)
Ln 2, Col 72 Spaces: 2 UTF-8 LF JSON with Comments

```
counter: not available
result: not available
aux.length: not available
aux[aux.length-1]: not availab…
sentinel: not available
i: not available
j: not available
x: not available
```

```
All Exceptions
Uncaught Exceptions
asd.js src 66
breakpoint.js src 20
logpoint.js src 21
variables.js src 22
variables.js src 25
```

```
{
  // Use IntelliSense to learn about possible attributes. (ctrl + space)
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    // Config for variables.js
    {
      "type": "node",
      "request": "launch",
      "name": "Launch variables.js",
      "skipFiles": [
        "<node_internals>/**"
      ],
      "program": "${workspaceFolder}/src/variables.js"
    }

    // Config for logpoint.js
    {
      "type": "node",
      "request": "launch",
      "name": "Launch logpoint.js",
      "skipFiles": [
        "<node_internals>/**"
      ],
      "program": "${workspaceFolder}/src/logpoint.js"
    }

    // Config for breakpoint.js
    {
      "type": "node",
      "request": "launch",
      "name": "Launch breakpoints.js",
      "skipFiles": [
```

5.4 Node Debug



Node Debug

Azure Pipelines never built

This extension is bundled with Visual Studio Code and together with **Node Debug (legacy)** forms the Node.js debugging experience.

Node Debug is the debugger for Node.js versions >= 8.0.

See a general overview of debugging in VS Code here.

Documentation for Node.js specific debugging can be found here.

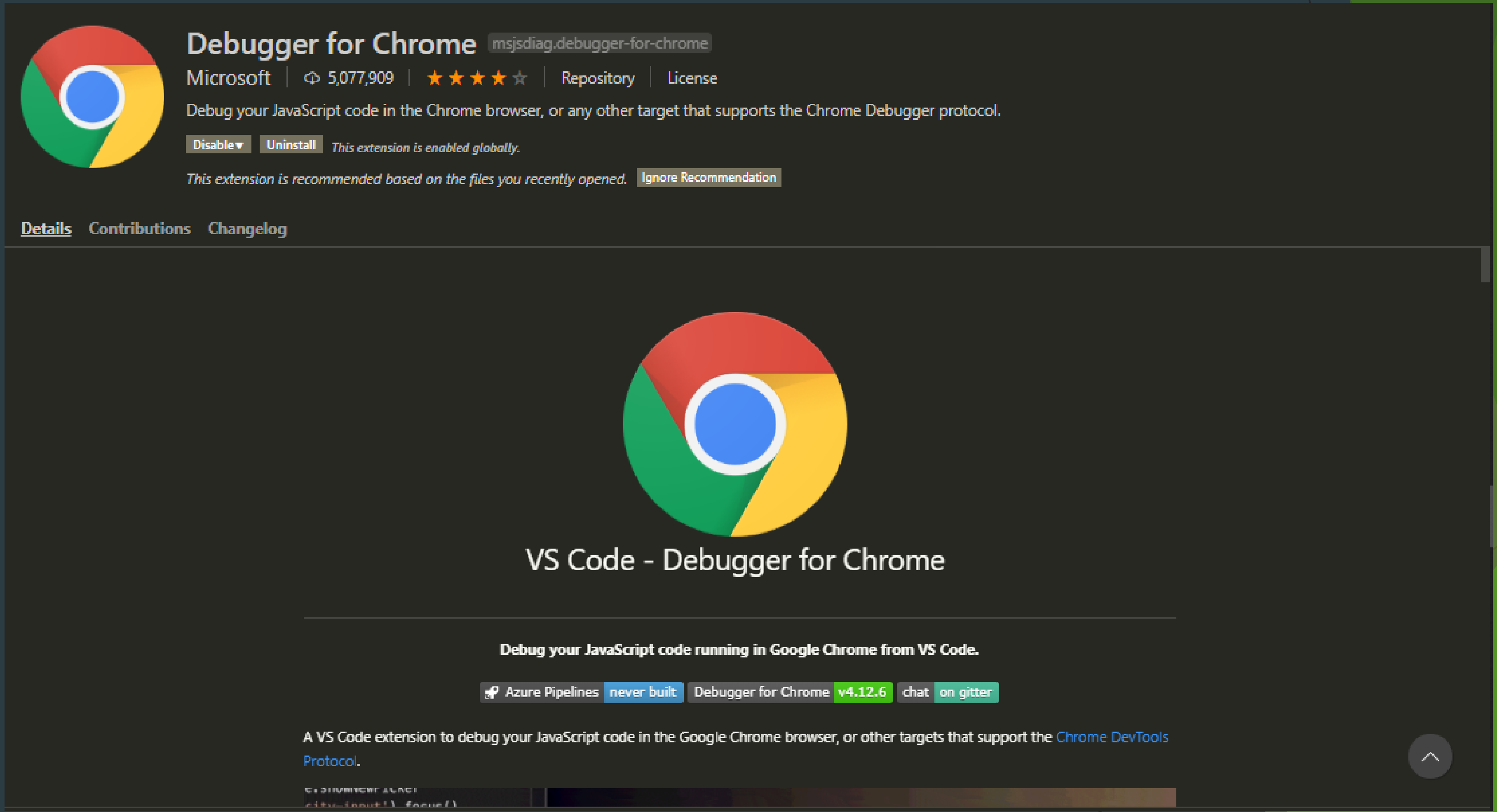
Please submit bugs and feature requests to the VS Code repository.

License

Copyright (c) Microsoft Corporation. All rights reserved.

Licensed under the MIT License.

5.5 Debugger for Chrome



Debugger for Chrome msjsdiag.debugger-for-chrome

Microsoft | 5,077,909 | ★★★★☆ | Repository | License

Debug your JavaScript code in the Chrome browser, or any other target that supports the Chrome Debugger protocol.

Disable ▾ Uninstall *This extension is enabled globally.*

This extension is recommended based on the files you recently opened. Ignore Recommendation

Details Contributions Changelog

VS Code - Debugger for Chrome

Debug your JavaScript code running in Google Chrome from VS Code.

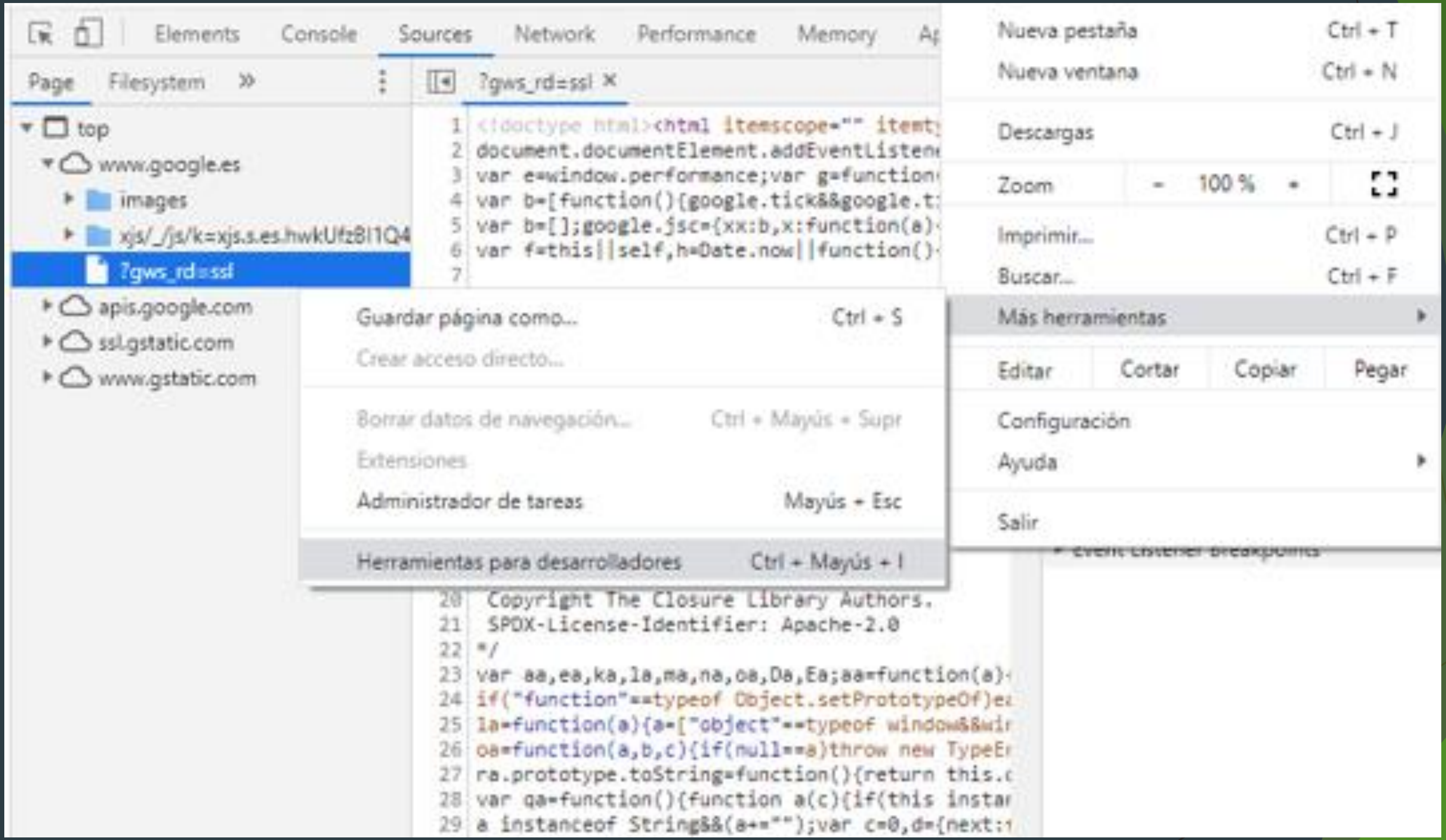
Azure Pipelines never built | Debugger for Chrome v4.12.6 | chat on gitter

A VS Code extension to debug your JavaScript code in the Google Chrome browser, or other targets that support the Chrome DevTools Protocol.

6. Chrome and DevTools

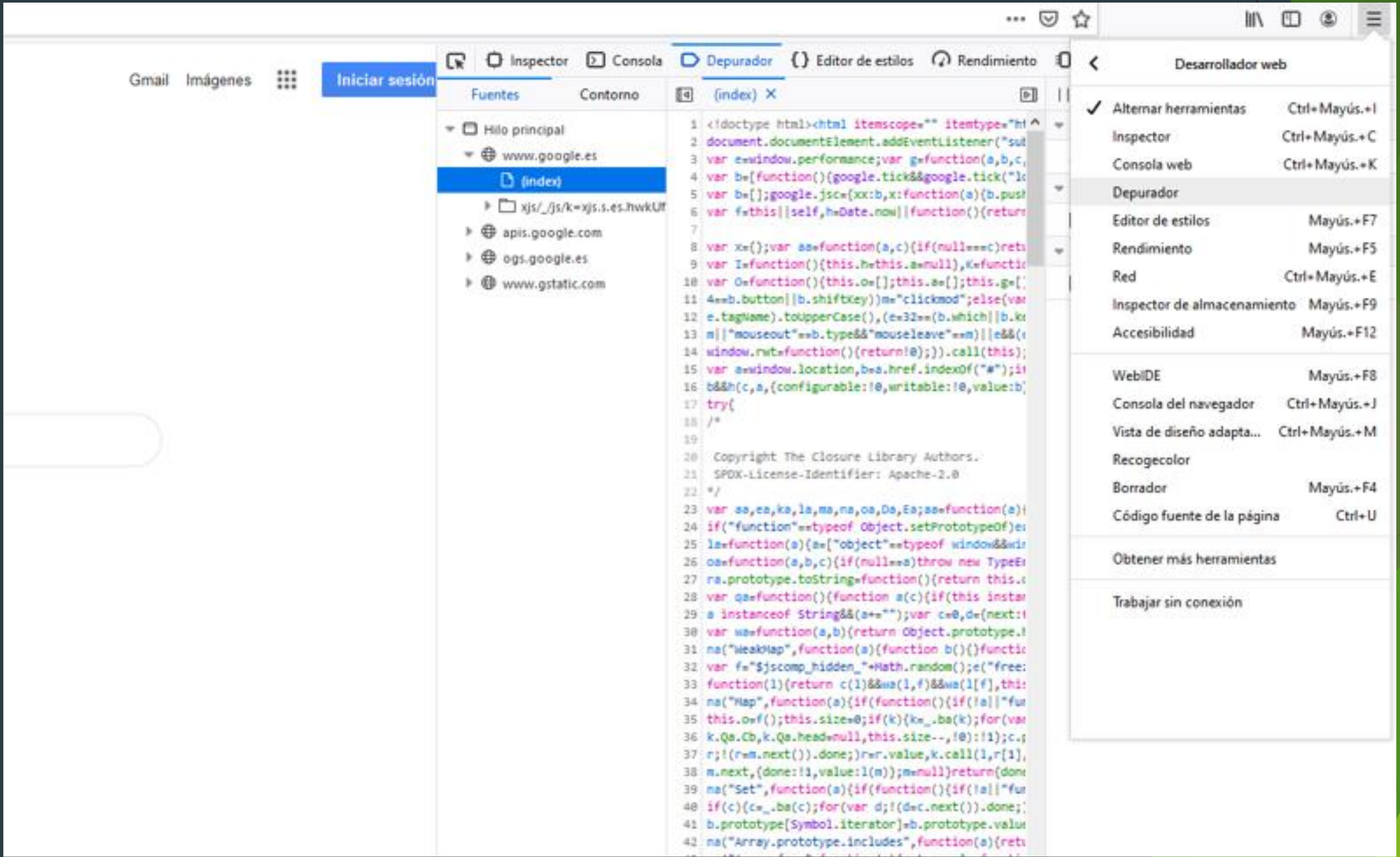
- Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser. DevTools can help you edit pages on-the-fly and diagnose problems quickly, which ultimately helps you build better websites, faster.

6.1 Chrome and Dev Tools



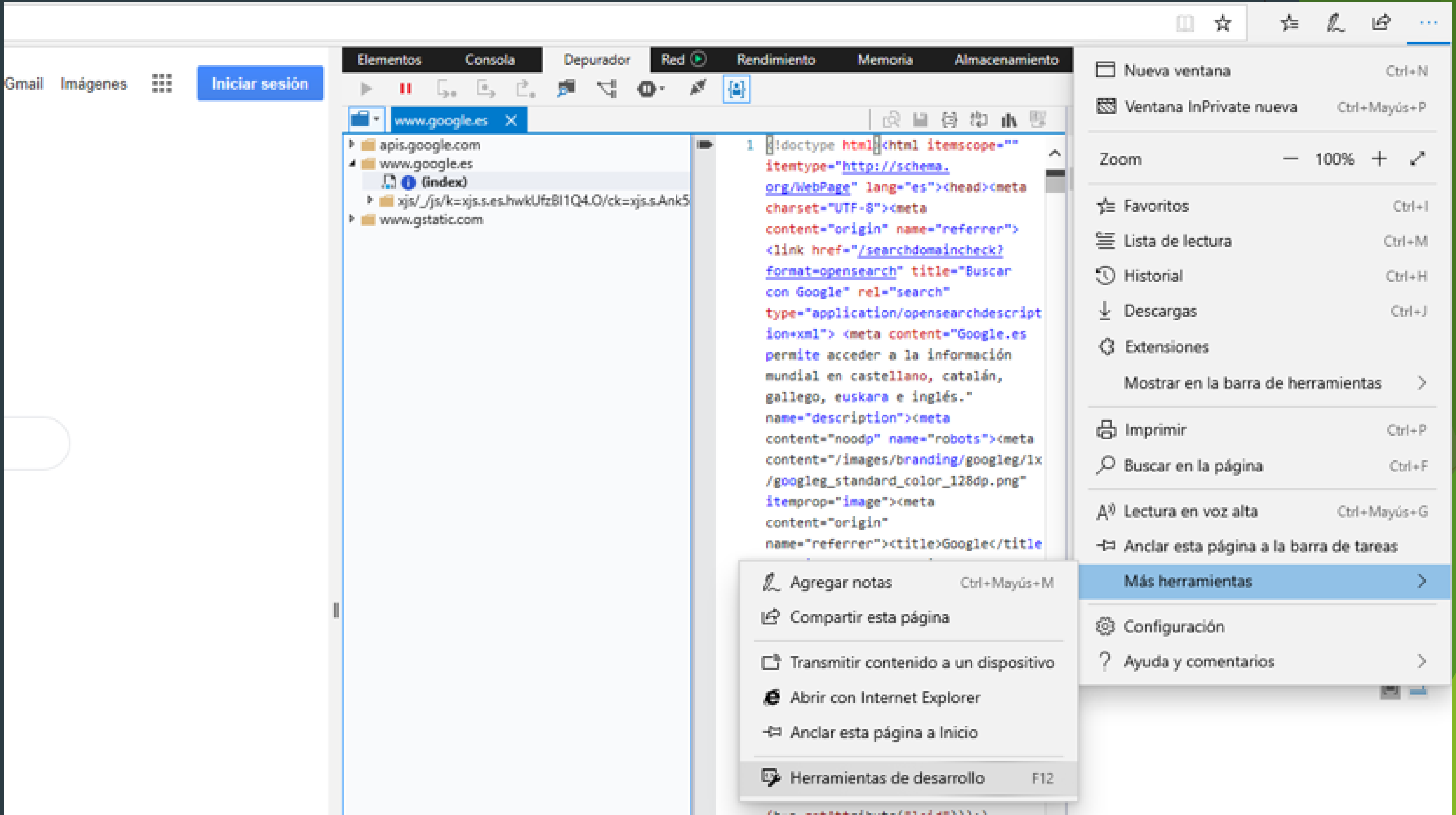
```
<!doctype html><html itemscope="" itemt
document.documentElement.addEventListen
var e=window.performance;var g=function
var b=[function(){google.tick&&google.t
var b=[];google.jsc={xx:b,x:function(a)
var f=this||self,h=Date.now||function()

 Copyright The Closure Library Authors.
 SPDX-License-Identifier: Apache-2.0
*/
var aa,ea,ka,la,ma,na,oa,Da,Ea;aa=function(a)
if("function"==typeof Object.setPrototypeOf)e
la=function(a){a=["object"==typeof window&&wi
oa=function(a,b,c){if(null==a)throw new TypeE
ra.prototype.toString=function(){return this.
var qa=function(){function a(c){if(this instar
a instanceof String&&(a+="");var c=0,d={next:
```

6.2 Firefox



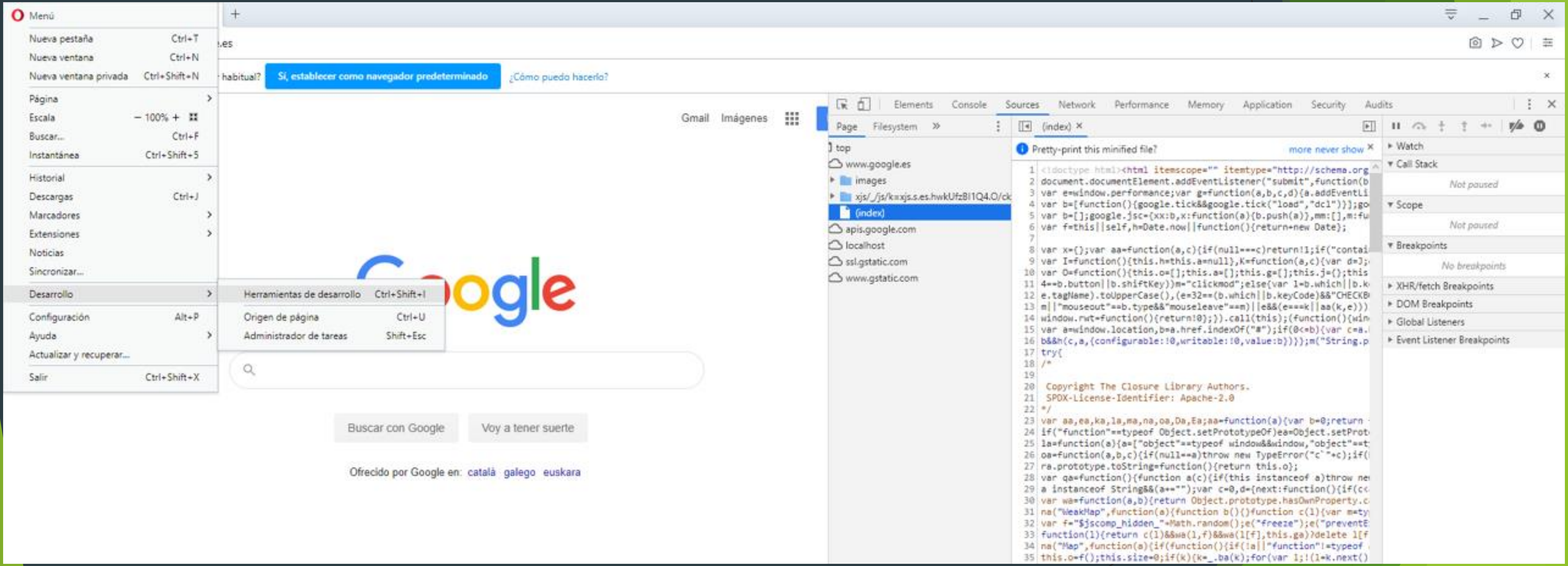
```
<!doctype html><html itemscope="" itemtype="ht
document.documentElement.addEventListener("sub
var e=window.performance;var g=function(a,b,c,
var b=[function(){google.tick&&google.tick("lo
var b=[];google.jsc={xx:b,x:function(a){b.push
var f=this||self,h=Date.now||function(){return

var X={};var aa=function(a,c){if(null===c)retu
var I=function(){this.h=this.a=null},K=functio
var O=function(){this.o=[];this.a=[];this.g=[
4==b.button||b.shiftKey))m="clickmod";else(var
e.tagName).toUpperCase(),(e=32==(b.which||b.ke
m||"mouseout"==b.type&&"mouseleave"==m)||e&&(
window.rwt=function(){return!0};}).call(this);
var a=window.location,b=a.href.indexOf("#");if
b&&h(c,a,{configurable:!0,writable:!0,value:b}
try{
/*

 Copyright The Closure Library Authors.
 SPDX-License-Identifier: Apache-2.0
*/
var aa,ea,ka,la,ma,na,oa,Da,Ea;aa=function(a){
if("function"==typeof Object.setPrototypeOf)ea
la=function(a){a=["object"==typeof window&&win
oa=function(a,b,c){if(null==a)throw new TypeEr
ra.prototype.toString=function(){return this.a
var qa=function(){function a(c){if(this instan
a instanceof String&&(a+="");var c=0,d={next:f
var wa=function(a,b){return Object.prototype.h
na("WeakMap",function(a){function b(){}functio
var f="$jscomp_hidden_"+Math.random();e("freez
function(l){return c(l)&&wa(l,f)&&wa(l[f],this
na("Map",function(a){if(function(){if(!a||"fun
this.o=f();this.size=0;if(k){k=_.ba(k);for(var
k.Qa.Cb,k.Qa.head=null,this.size--,!0):!1};c.p
r;!(r=m.next()).done;)r=r.value,k.call(l,r[1],
m.next,{done:!1,value:l(m)};m=null}return{done
na("Set",function(a){if(function(){if(!a||"fun
if(c){c=_.ba(c);for(var d;!(d=c.next()).done;)
b.prototype[Symbol.iterator]=b.prototype.value
na("Array.prototype.includes",function(a){retu
```

6.3 Microsoft Edge


Gmail
Imágenes
Iniciar sesión
Elementos
Consola
Depurador
Red
Rendimiento
Memoria
Almacenamiento
www.google.es
apis.google.com
www.google.es
(index)
xjs/_/js/k=xjs.s.es.hwkUfzBI1Q4.O/ck=xjs.s.Ank5
www.gstatic.com
Nueva ventana Ctrl+N
Ventana InPrivate nueva Ctrl+Mayús+P
Zoom — 100% +
Favoritos Ctrl+I
Lista de lectura Ctrl+M
Historial Ctrl+H
Descargas Ctrl+J
Extensiones
Mostrar en la barra de herramientas
Imprimir Ctrl+P
Buscar en la página Ctrl+F
Lectura en voz alta Ctrl+Mayús+G
Anclar esta página a la barra de tareas
Más herramientas
Configuración
Ayuda y comentarios
Agregar notas Ctrl+Mayús+M
Compartir esta página
Transmitir contenido a un dispositivo
Abrir con Internet Explorer
Anclar esta página a Inicio
Herramientas de desarrollo F12

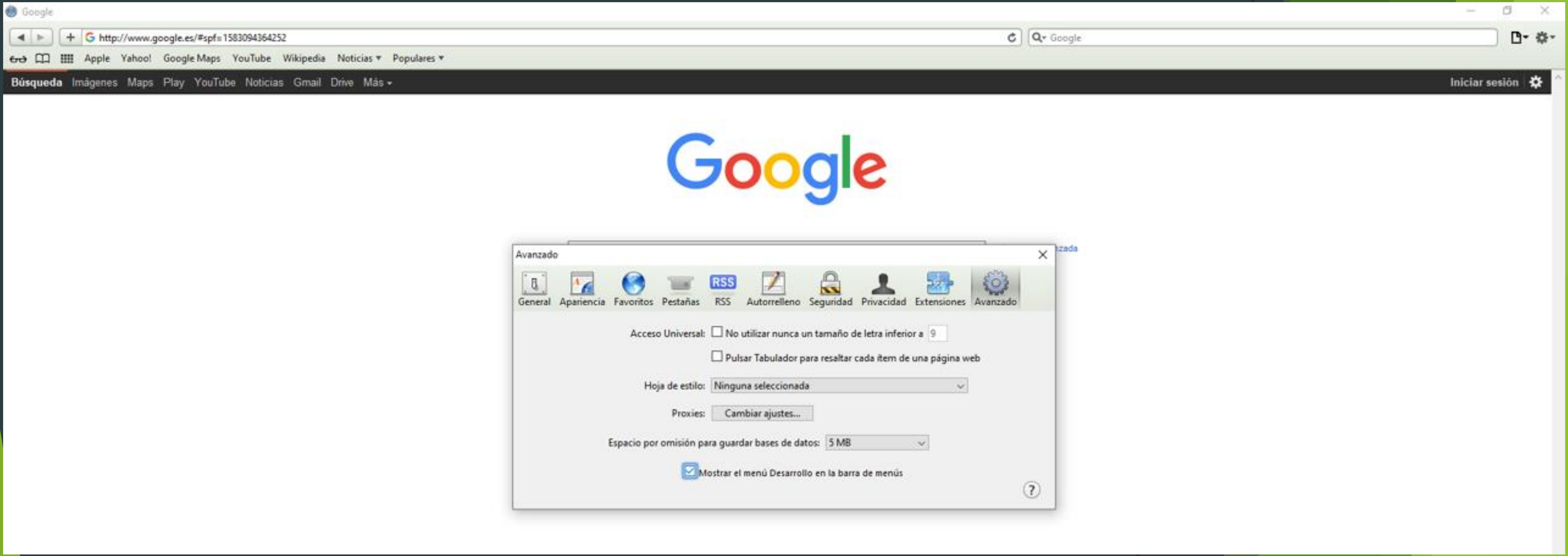
```
<!doctype html><html itemscope=""
itemtype="http://schema.
org/WebPage" lang="es"><head><meta
charset="UTF-8"><meta
content="origin" name="referrer">
<link href="/searchdomaincheck?
format=opensearch" title="Buscar
con Google" rel="search"
type="application/opensearchdescript
ion+xml"> <meta content="Google.es
permite acceder a la información
mundial en castellano, catalán,
gallego, euskara e inglés."
name="description"><meta
content="noodp" name="robots"><meta
content="/images/branding/googleg/1x
/googleg_standard_color_128dp.png"
itemprop="image"><meta
content="origin"
name="referrer"><title>Google</title
```

6.4 Opera



```
<!doctype html><html itemscope="" itemtype="http://schema.org
document.documentElement.addEventListener("submit",function(b
var e=window.performance;var g=function(a,b,c,d){a.addEventLi
var b=[function(){google.tick&&google.tick("load","dcl")}];go
var b=[];google.jsc={xx:b,x:function(a){b.push(a)},mm:[],m:fu
var f=this||self,h=Date.now||function(){return+new Date};

var x={};var aa=function(a,c){if(null===c)return!1;if("contai
var I=function(){this.h=this.a=null},K=function(a,c){var d=J;
var O=function(){this.o=[];this.a=[];this.g=[];this.j={};this
4==b.button||b.shiftKey))m="clickmod";else{var l=b.which||b.k
e.tagName).toUpperCase(),(e=32==(b.which||b.keyCode)&&"CHECKB
m||"mouseout"==b.type&&"mouseleave"==m)||e&&(e===k||aa(k,e)))
window.rwt=function(){return!0};}).call(this);(function(){win
var a=window.location,b=a.href.indexOf("#");if(0<=b){var c=a.
b&&h(c,a,{configurable:!0,writable:!0,value:b})}});m("String.p
try{
/*

 Copyright The Closure Library Authors.
 SPDX-License-Identifier: Apache-2.0
*/
var aa,ea,ka,la,ma,na,oa,Da,Ea;aa=function(a){var b=0;return
if("function"==typeof Object.setPrototypeOf)ea=Object.setProt
la=function(a){a=["object"==typeof window&&window,"object"==t
oa=function(a,b,c){if(null==a)throw new TypeError("c`"+c);if(
ra.prototype.toString=function(){return this.o};
var qa=function(){function a(c){if(this instanceof a)throw ne
a instanceof String&&(a+="");var c=0,d={next:function(){if(c<
var wa=function(a,b){return Object.prototype.hasOwnProperty.c
na("WeakMap",function(a){function b(){}function c(l){var m=ty
var f="$jscomp_hidden_"+Math.random();e("freeze");e("preventE
function(l){return c(l)&&wa(l,f)&&wa(l[f],this.ga)?delete l[f
na("Map",function(a){if(function(){if(!a||"function"!=typeof
this.o=f();this.size=0;if(k){k=_.ba(k);for(var l;!(l=k.next()
```

6.5 Safari



Avanzado

Acceso Universal: ☐ No utilizar nunca un tamaño de letra inferior a 9

☐ Pulsar Tabulador para resaltar cada ítem de una página web

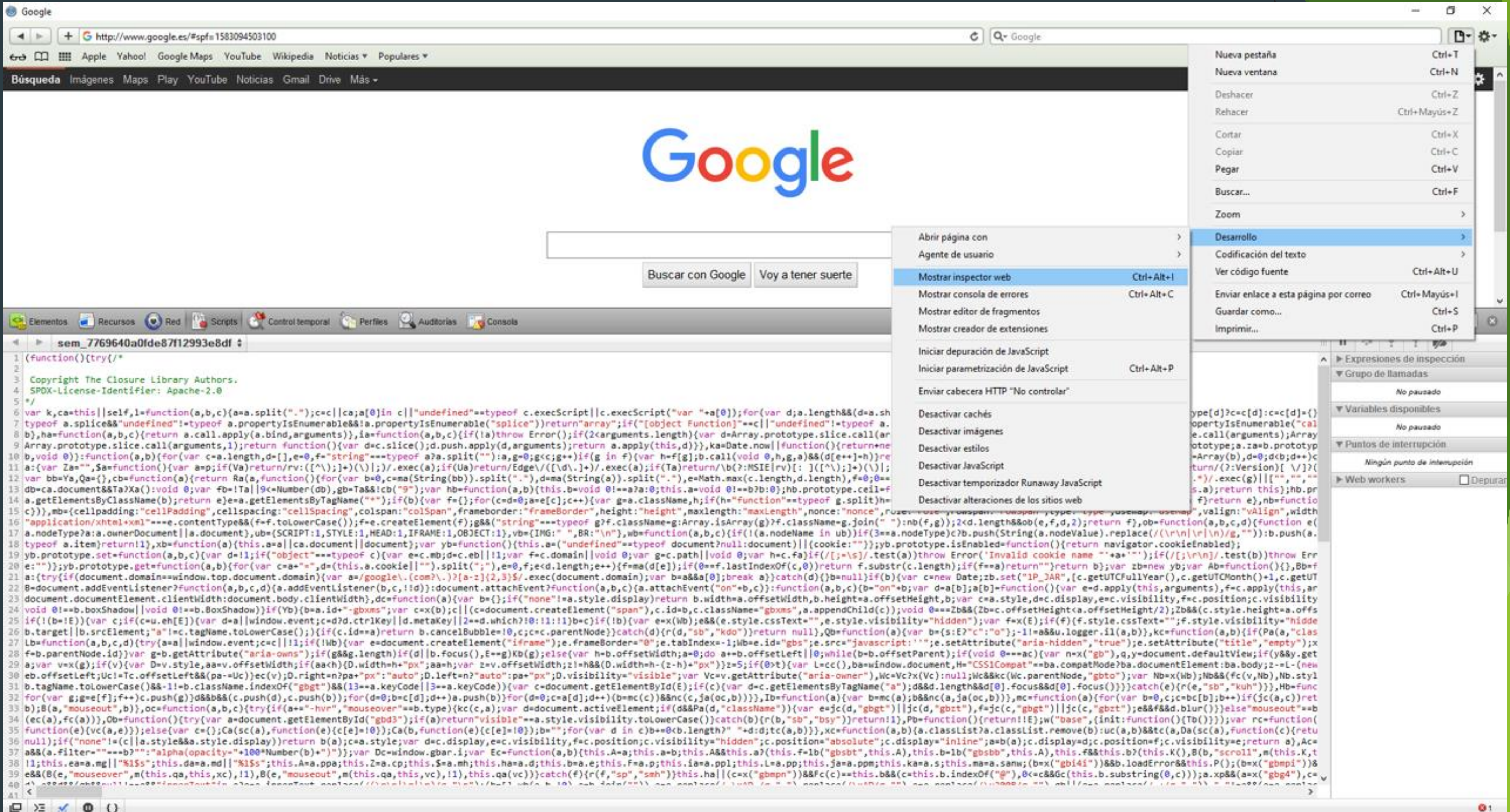
Hoja de estilo: Ninguna seleccionada

Proxies: Cambiar ajustes...

Espacio por omisión para guardar bases de datos: 5 MB

☑ Mostrar el menú Desarrollo en la barra de menús

6.5.2 Safari



```
(function(){try{/*

 Copyright The Closure Library Authors.
 SPDX-License-Identifier: Apache-2.0
*/
var k,ca=this||self,l=function(a,b,c){a=a.split(".");c=c||ca;a[0]in c||"undefined"==typeof c.execScript||c.execScript("var "+a[0]);for(var d;a.length&&(d=a.sh
```

Bibliography

- Debugger definition
- Strategies
- Visual Studio Code
- Using Nodejs in Visual Studio Code

- Debugging a chat in VSC
- Tutorial and Install Nodejs in VSC
- Node.js
- Google Dev Tools

Github

- This presentation with all the examples used on it are available in our public repository at github:

https://github.com/ULL-ESIT-INF-PAI-2019-2020/2019-2020-pai-trabajo-debugging-adrian-rodriguez-sergio-tabares

Contact

- Sergio Tabares Hernández: alu0101124896@ull.edu.es
- Adrián Epifanio Rodríguez Hernández: alu0101158280@ull.edu.es

Thanks you for your attention, if you have any question please let us to know it.