Debugging JavaScript and TypeScript

Understanding the Problem

José Gregorio
DortaLuis

alu0101414676@ull.edu.es
jose.dorta.31@ull.edu.es

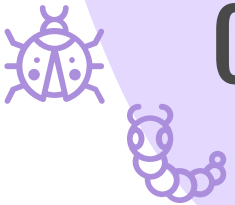
Muhammad
Campos Preira

alu0101434025@ull.edu.es
muhammad.campos.15@ull.edu.es

Table of Contents

Bugs

What Are Computer Bugs?



Early bugs detection

Why is it so important?

Why is it so important?

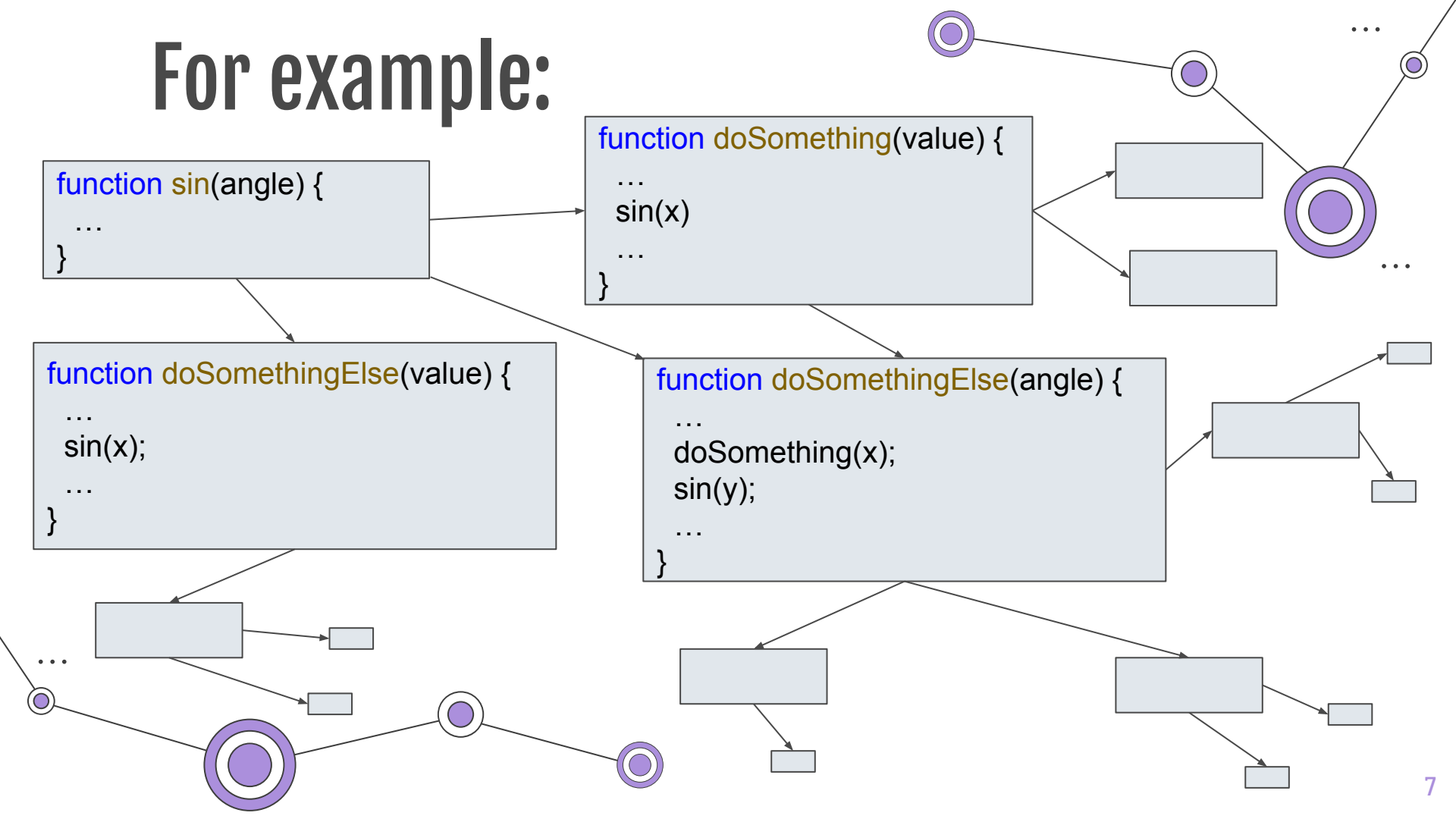
Reduction in development time

Cost reduction

Makes debugging easier

Improves the reputation of a company

For example:


function sin(angle) {
…
}
function doSomething(value) {
…
sin(x)
…
}
function doSomethingElse(value) {
…
sin(x);
…
}
function doSomethingElse(angle) {
…
doSomething(x);
sin(y);
…
}

Improve companies reputation?

Hogwarts legacy

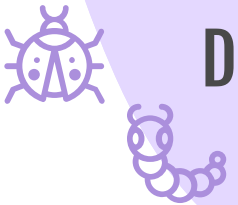
Assassin's Creed
Unity

Battlefield 4

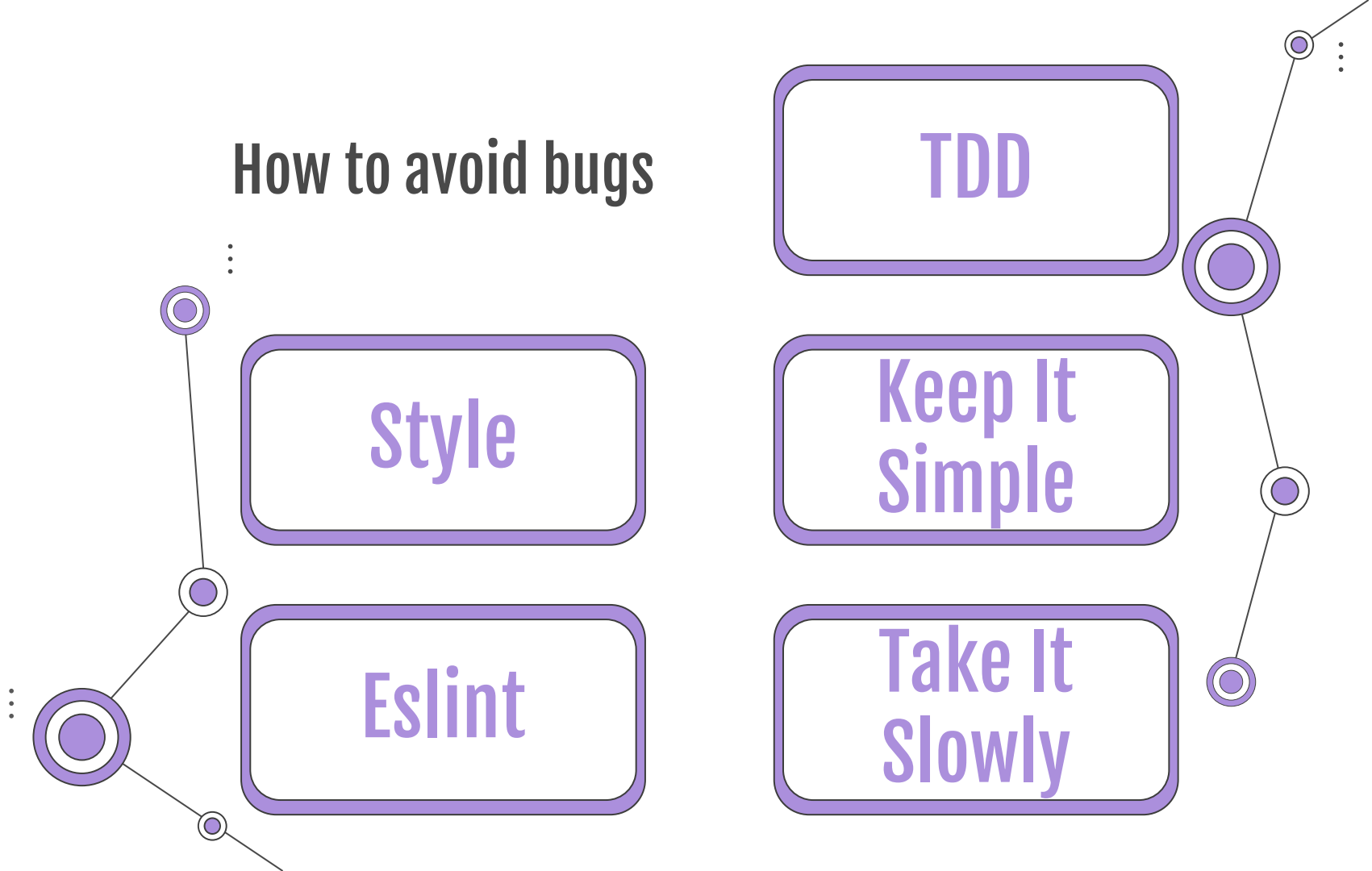
ARIANE 5G de 1996

How to avoid bugs

Different ways to reduce bugs.

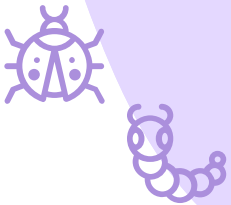


How to avoid bugs



Debugging

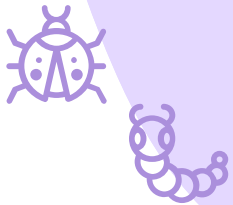
What is it?



Debugging

- Is the process of finding and fixing errors (also known as bugs) in a computer program's
- Debugging is done to ensure that the program runs correctly and without errors.
- Debugging is typically done using software tools such as IDEs.
- Debugging is a critical part of the software development process

Tools and methods to debug



Logging, Unit testing...

Logging

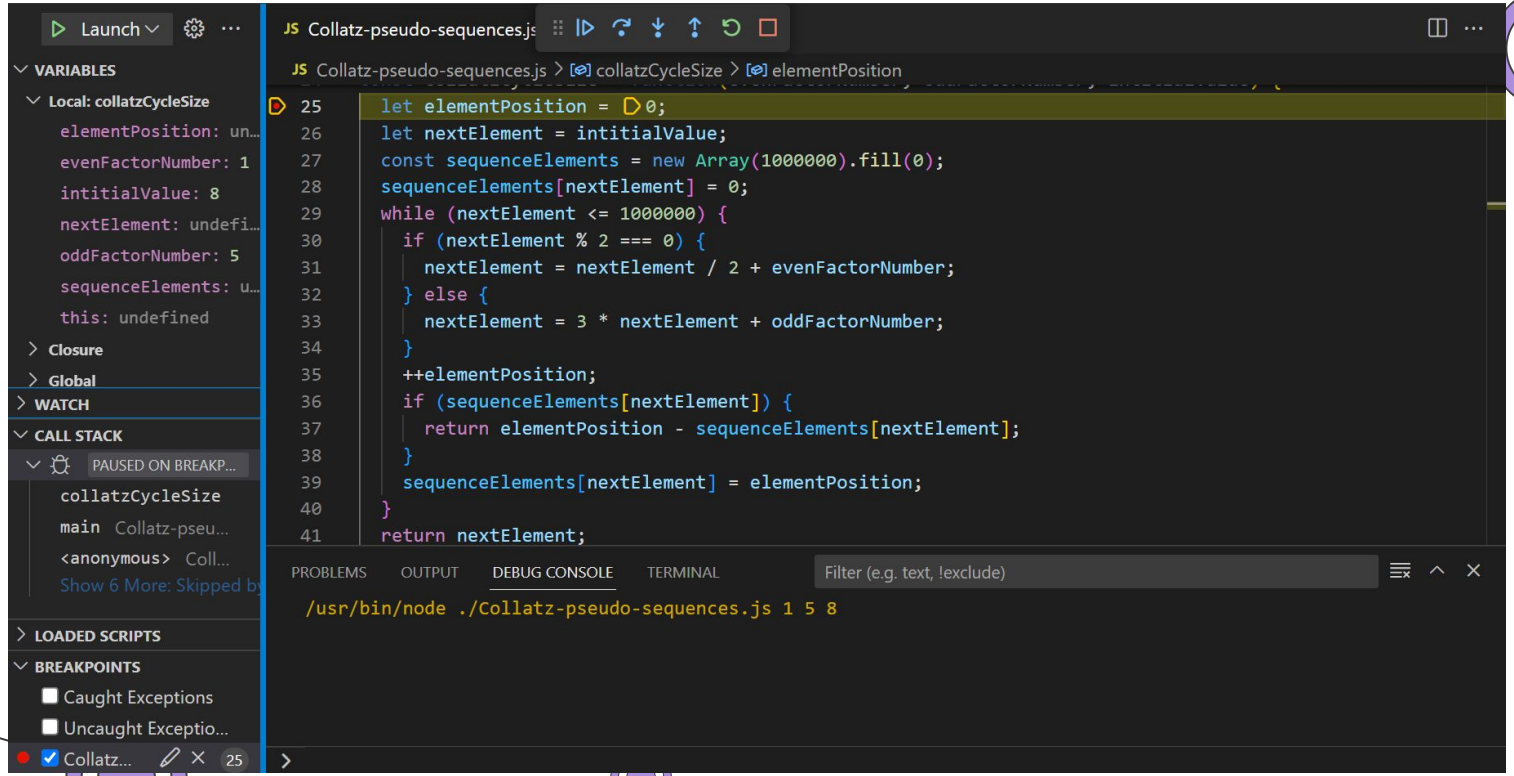
```
const collatzCycleSize = function(evenFactorNumber, oddFactorNumber, intitialValue) {
  console.log('intitialValue: ', intitialValue);
  console.log('evenFactorNumber: ', evenFactorNumber);
  console.log('oddFactorNumber: ', oddFactorNumber);
  let elementPosition = 0;
  let nextElement = intitialValue;
  const sequenceElements = new Array(1000000).fill(0);
  sequenceElements[nextElement] = 0;
  while (nextElement <= 1000000) {
    if (nextElement % 2 === 0) {
      nextElement = nextElement / 2 + evenFactorNumber;
      console.log('nextElement: ', nextElement);
    } else {
      nextElement = 3 * nextElement + oddFactorNumber;
      console.log('nextElement: ', nextElement);
```

Unit testing

```
describe('collatzCycleSize function', function() {
    it('should return 3 for 1, 5, 8', function() {
        assert.equal(collatzCycleSize(1, 5, 8), 3);
    });
    it('should return 1 for 0, 5, 0', function() {
        assert.equal(collatzCycleSize(0, 5, 0), 1);
    });
    it('should return 1 for 10, 11, 3', function() {
        assert.equal(collatzCycleSize(10, 11, 3), 1);
    });
    it('should return 35 for 7, 3, 6', function() {
        assert.equal(collatzCycleSize(7, 3, 6), 35);
    });
    it('should return 1501002 for 1, 999, 1000000', function() {
        assert.equal(collatzCycleSize(1, 999, 1000000), 1501002);
    });
    it('should return 490 for 433, 805, 215476', function() {
        assert.equal(collatzCycleSize(433, 805, 215476), 490);
    });
});
```

Pair programing

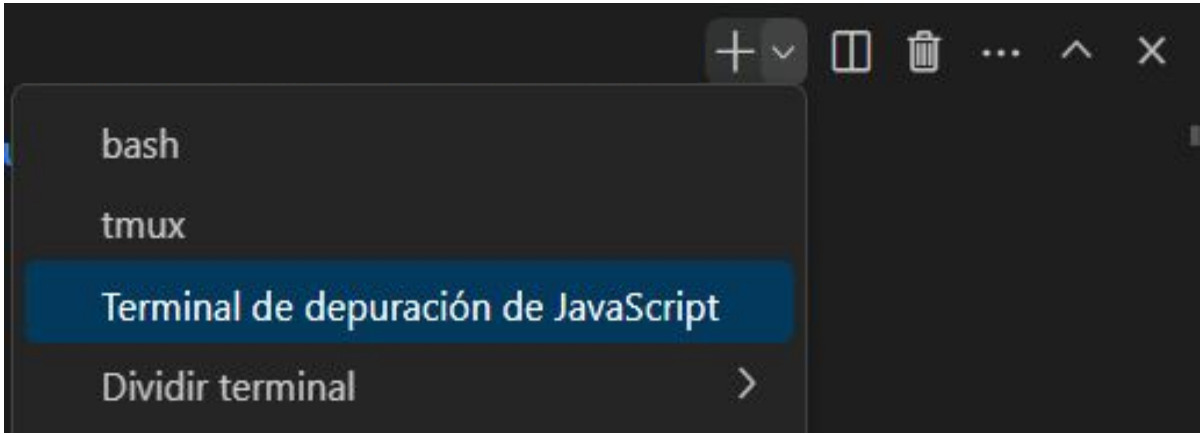
Debugger


Launch
VARIABLES
Local: collatzCycleSize
elementPosition: un...
evenFactorNumber: 1
intitialValue: 8
nextElement: undefi...
oddFactorNumber: 5
sequenceElements: u...
this: undefined
Closure
Global
WATCH
CALL STACK
PAUSED ON BREAKP...
collatzCycleSize
main Collatz-pseu...
<anonymous> Coll...
Show 6 More: Skipped by
LOADED SCRIPTS
BREAKPOINTS
Caught Exceptions
Uncaught Exceptio...
Collatz... 25
Collatz-pseudo-sequences.js
Collatz-pseudo-sequences.js > collatzCycleSize > elementPosition
25 let elementPosition = 0;
26 let nextElement = intitialValue;
27 const sequenceElements = new Array(1000000).fill(0);
28 sequenceElements[nextElement] = 0;
29 while (nextElement <= 1000000) {
30 if (nextElement % 2 === 0) {
31 nextElement = nextElement / 2 + evenFactorNumber;
32 } else {
33 nextElement = 3 * nextElement + oddFactorNumber;
34 }
35 ++elementPosition;
36 if (sequenceElements[nextElement]) {
37 return elementPosition - sequenceElements[nextElement];
38 }
39 sequenceElements[nextElement] = elementPosition;
40 }
41 return nextElement;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Filter (e.g. text, !exclude)
/usr/bin/node ./Collatz-pseudo-sequences.js 1 5 8

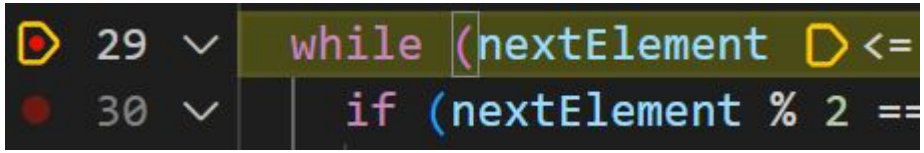
Debugging in VS Code

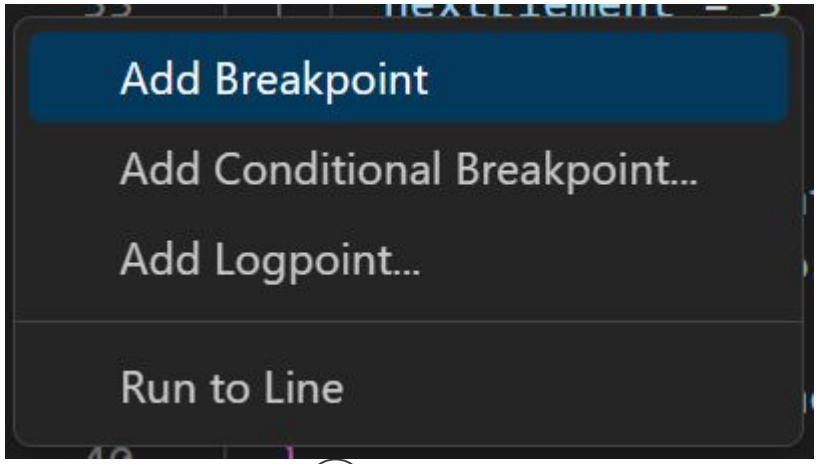
Configuration, tools, breakpoints...

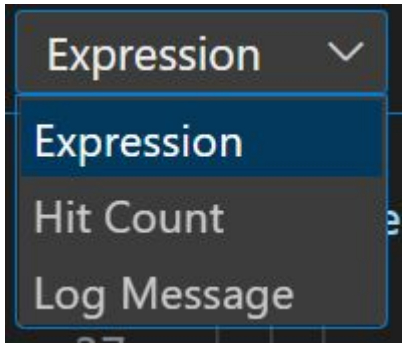
JavaScript debug terminal



Breakpoints



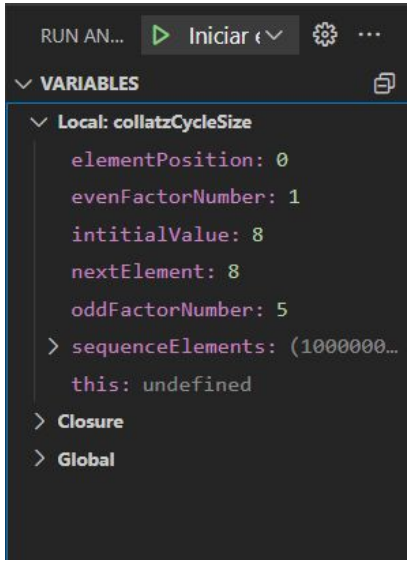


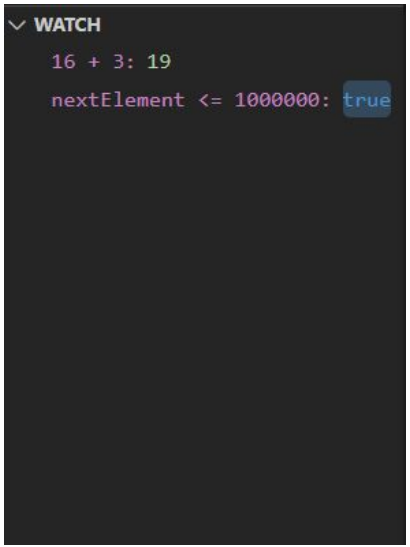


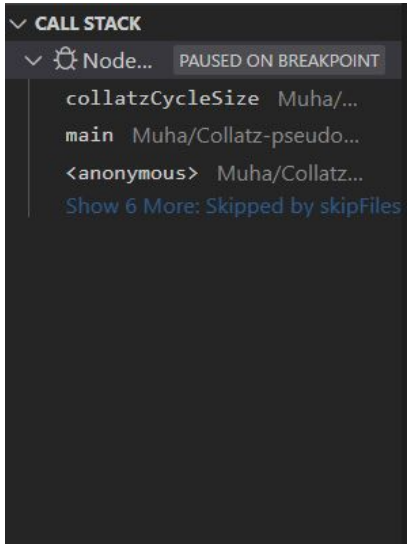
Buttons

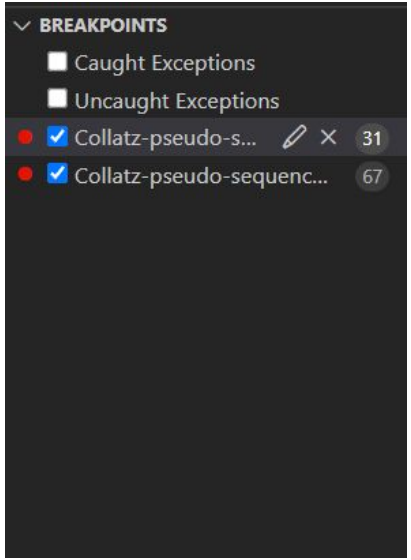
- Disconnect
- Restart
- Step out
- Step into
- Step over
- Continue

Side panel

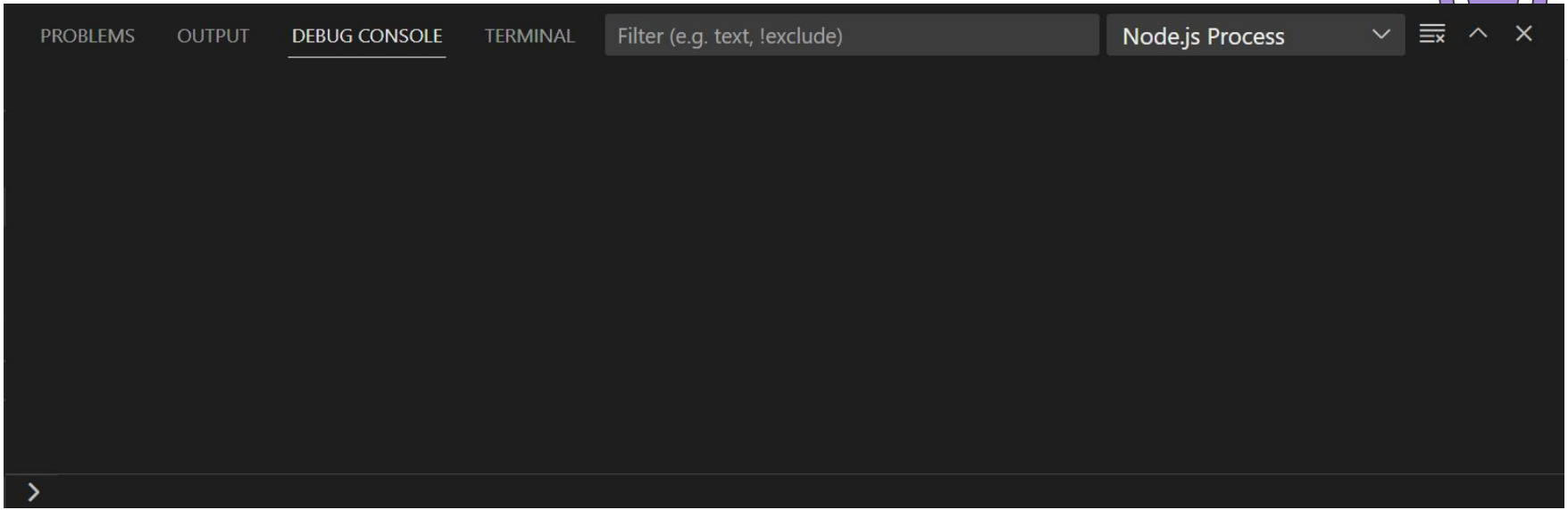

RUN AN...
Iniciar
VARIABLES
Local: collatzCycleSize
elementPosition: 0
evenFactorNumber: 1
intitialValue: 8
nextElement: 8
oddFactorNumber: 5
sequenceElements: (1000000…
this: undefined
Closure
Global


WATCH
16 + 3: 19
nextElement <= 1000000: true

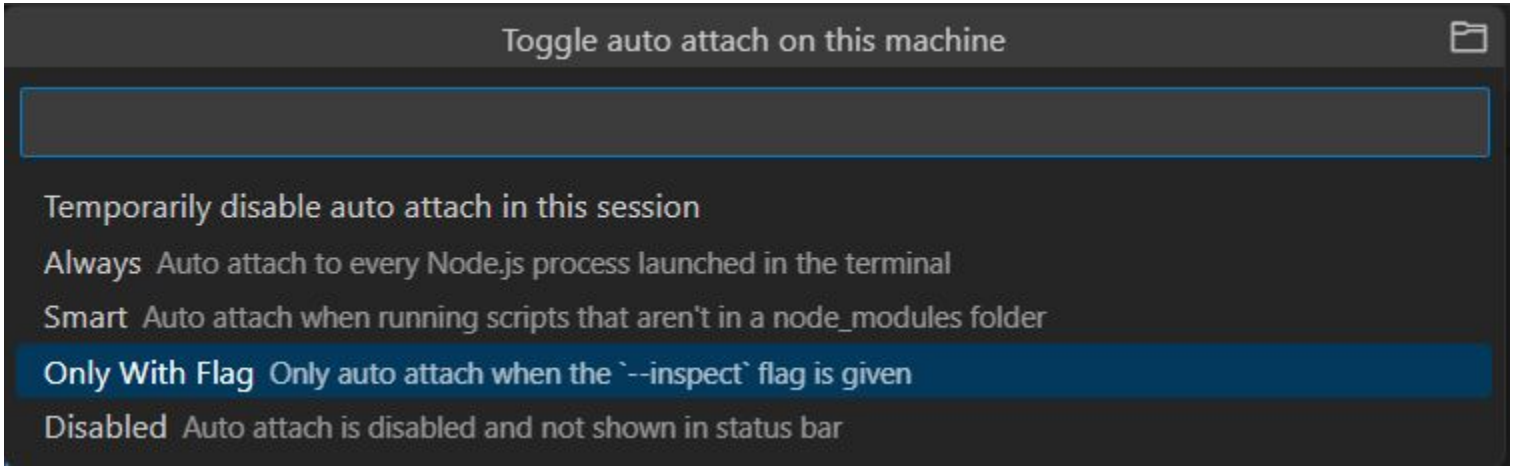

CALL STACK
Node...
PAUSED ON BREAKPOINT
collatzCycleSize Muha/...
main Muha/Collatz-pseudo...
<anonymous> Muha/Collatz...
Show 6 More: Skipped by skipFiles


BREAKPOINTS
Caught Exceptions
Uncaught Exceptions
Collatz-pseudo-s...
31
Collatz-pseudo-sequenc...
67

Debug console



Auto attach


Toggle auto attach on this machine
Temporarily disable auto attach in this session
Always Auto attach to every Node.js process launched in the terminal
Smart Auto attach when running scripts that aren't in a node_modules folder
Only With Flag Only auto attach when the `--inspect` flag is given
Disabled Auto attach is disabled and not shown in status bar

launch.json

.vscode > {} launch.json > ...

```jsonc
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "node",
            "request": "launch",
            "name": "Launch Program",
            "skipFiles": [
                "<node_internals>/**"
            ],
            "program": "${workspaceFolder}/Collatz-pseudo-sequences.js",
            "args": ["1", "5", "8"]
        }
    ]
}
```

Browser debugging

References

- https://code.visualstudio.com/docs/nodejs/nodejs-debugging
- https://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/verificacion/testing-temprano
- https://www.testingit.com.mx/blog/bugs-en-software
- https://www.guardadorapido.com/los-5-bugs-mas-famosos-de-los-videojuegos/
- https://as.com/meristation/2015/11/16/reportajes/1447657200_150690.html
- https://www.quora.com/What-are-good-ways-to-avoid-bugs-while-programming
- https://enou.co/blog/how-to-reduce-bugs-in-software/
- https://simpleprogrammer.com/reduce-software-bugs-quality-code/

Thanks!