



Ajuntament
de Barcelona

Debugger

Comenzar a depurar en Chrome

IT Academy

Mayo de 2019



Comenzando a depurar en CSS



Modificar elementos CSS

Vamos a la pestaña ELEMENTS y modificamos los estilos CSS. **Si recargamos la pantalla perderemos todos los cambios.**

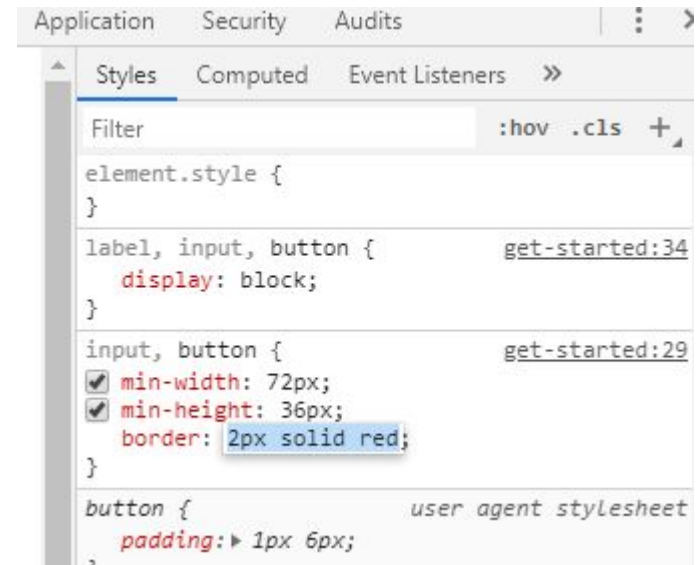
Demo: Get Started Debugging JavaScript with Chrome DevTools

Number 1

Number 2

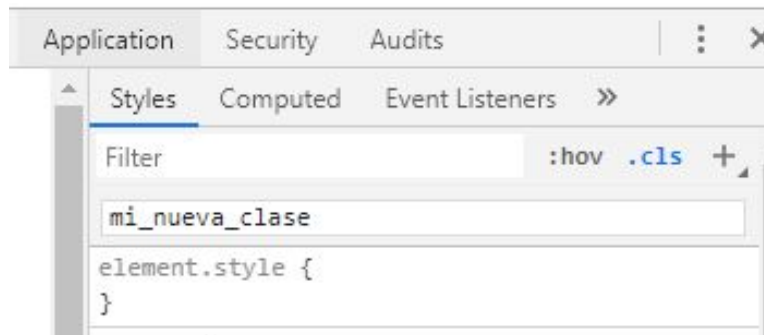
Add Number 1 and Number 2

3 + 2 = 32





Añadir clases al CSS



```
<label for="num2">Number 2</label>
<input placeholder="Number 2" id="num2">
<button class="mi_nueva_clase">Add Number 1 and
<p>3 + 2 = 32</p>
<script src="get-started.js"></script>
</body>
</html>
```



Comenzando a depurar Javascript



Paso 1: Reproducir el error

<https://googlechrome.github.io/devtools-samples/debug-js/get-started>

El primer paso para la depuración siempre es encontrar una serie de acciones que reproduzcan un error de manera uniforme.

A screenshot of a web application interface. It features two input fields labeled 'Number 1' and 'Number 2'. The first field contains the value '6' and the second contains '2'. Below these fields is a button labeled 'Add Number 1 and Number 2'. At the bottom of the interface, the result of the addition is displayed as '6 + 2 = 62', which is a clear error as the correct result should be 8.



Paso 2: Activamos la consola (DevTools)

Pulsando F12 activaremos la consola de depuración de Chrome, la **DevTools**. Luego haremos clic en la pestaña **Sources**.

The screenshot displays the Chrome DevTools interface. On the left, a demo application titled "Demo: Get Started Debugging JavaScript with Chrome DevTools" is visible. It contains two input fields labeled "Number 1" and "Number 2", with values "6" and "2" respectively. Below these is a button labeled "Add Number 1 and Number 2". The result of the calculation, "6 + 2 = 62", is shown at the bottom of the demo area.

The DevTools interface is open, with the "Sources" tab selected and highlighted by a red box. The "Page" tab is also visible. The "Sources" panel shows a file tree on the left with "get-started.js" selected and highlighted by a red box. The main pane displays the JavaScript code for "get-started.js", which includes functions for handling input, updating the label, and calculating the sum of two numbers. The code is as follows:

```
15 if (inputsAreEmpty()) {
16   label.textContent = 'Error: one or both inputs are empty.';
17   return;
18 }
19 updateLabel();
20 }
21 function inputsAreEmpty() {
22   if (getNumber1() === '' || getNumber2() === '') {
23     return true;
24   } else {
25     return false;
26   }
27 }
28 function updateLabel() {
29   var addend1 = getNumber1();
30   var addend2 = getNumber2();
31   var sum = addend1 + addend2;
32   label.textContent = addend1 + ' + ' + addend2 + ' = ' + sum;
33 }
34 function getNumber1() {
35   return inputs[0].value;
36 }
37 function getNumber2() {
38   return inputs[1].value;
39 }
40 var inputs = document.querySelectorAll('input');
41 var label = document.querySelector('p');
42 var button = document.querySelector('button');
43 button.addEventListener('click', onClick);
44
```



Paso 3: Pausar el código - con BreakPoints

Pausa el código con un punto de interrupción

index.html hello.js x

```
1 function hello(name) { name = "John"
2   let phrase = `Hello, ${name}!`; phrase = "Hello, John!"
3
4   say(phrase);
5 }
6
7 function say(phrase) {
8   alert(`** ${phrase} **`);
9 }
10
```

watch expressions →

jump to the outer function →

current variables →

Paused on breakpoint

Watch + ↻

No Watch Expressions

Call Stack

- hello hello.js:4
- (anonymous) index.html:10

Scope

Local

- name: "John"
- phrase: "Hello, John!"
- this: Window

Global Window

{ } Line 4, Column 3



Paso 4: Recorre el código

Con el Debugger podremos:



Ejecuta la función entrando en la función



Ejecuta la función SIN entrar (o recorrer) la función



Le damos al Play, y continua la ejecución hasta llegar al final del programa o hasta encontrar otro breakpoint



Paso 5: Visualizar el valor de las variables

Paused in debugger

Demo: Get started
Debugging JavaScript with Chrome DevTools

Number 1
5

Number 2
2

Add Number 1 and Number 2

5 + 2 = 52

Elements Console Sources Network Performance Memory Application Security Audi

get-started content.js get-started.js x

```
22 if (getNumber1() === '' || getNumber2() === '') {
23   return true;
24 } else {
25   return false;
26 }
27 }
28 function updateLabel() {
29   var addend1 = getNumber1(); addend1 = "5"
30   var addend2 = getNumber2();
31   var sum = addend1 + addend2;
32   label.textContent = addend1 + ' + ' + addend2 + ' = ' +
33 }
34 function getNumber1() {
35   return inputs[0].value;
36 }
37 function getNumber2() {
38   return inputs[1].value;
39 }
40 var inputs = document.querySelectorAll('input');
41 var label = document.querySelector('p');
42 var button = document.querySelector('button');
43 button.addEventListener('click', onClick);
44
```

Debugger paused

Watch

No watch e

Call Stack

- updateLabel
- onClick

Scope

Local

- addend1: "5"
- addend2: undefined
- sum: undefined
- this: Window
 - alert: f alert()
 - applicationCache: /
 - atob: f atob()
 - blur: f ()



Paso 6: Expresiones supervisadas (Watch)

La pestaña **Watch** te permite supervisar los valores de las variables o expresiones a lo largo del tiempo.

```
22 if (getNumber1() === '' || getNumber2() === '') {
23   return true;
24 } else {
25   return false;
26 }
27 }
28 function updateLabel() {
29   var addend1 = getNumber1();
30   var addend2 = getNumber2();
31   var sum = addend1 + addend2;
32   label.textContent = addend1 + ' + ' + addend2 + ' = ' + sum;
33 }
34 function getNumber1() {
35   return inputs[0].value;
```

Debugger paused

▼ Watch

- addend1 == "3": true
- typeof sum: "string"

► Call Stack

▼ Scope

▼ Local

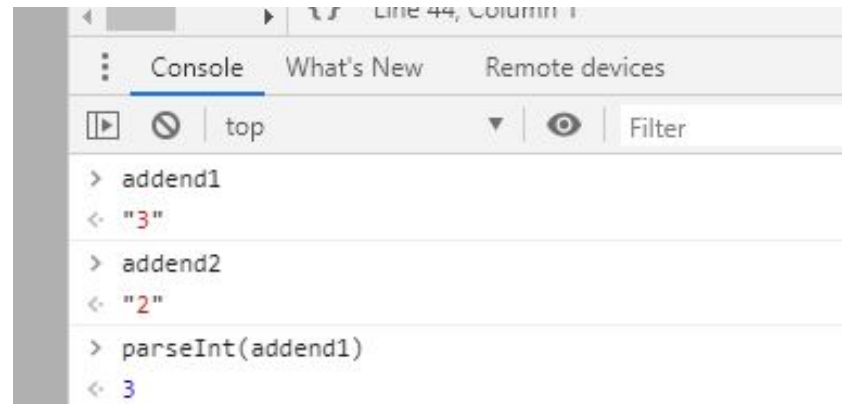
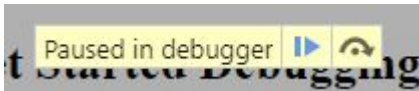
- addend1: "3"
- addend2: "2"
- sum: "32"



Paso 7: La consola

Utilizando la consola podremos acceder directamente a las variables o expresiones del código.

1. Activamos la consola con la tecla ESCAPE
2. Añadimos un breakpoint
3. Procedemos a escribir alguna expresión en la Consola





Barcelona
Activa



barcelona.cat/barcelonactiva