

Debug

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [¿Qué es “debug”?](#)
2. [Breakpoints](#)
3. [Interfaz](#)

1

¿Qué es
“debug”?



Llamamos **debuggear** al proceso de encontrar errores en nuestro código que impide su funcionamiento.



2 | Breakpoints



Un **breakpoint** es una pausa en nuestro código para entender en qué estado están nuestras variables.



Breakpoints

Breakpoint

Al hacer clic a la izquierda del número de línea, establecemos un breakpoint.

Contenido

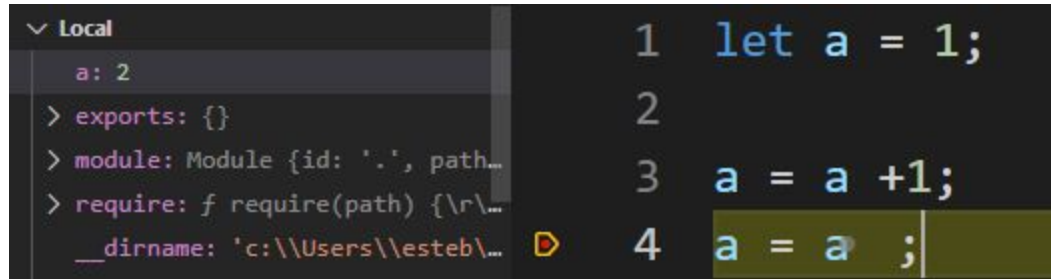
El contenido de nuestro código.



Número de línea

Nos indica el número de línea en el que está el código.

Control de variables



```
1 let a = 1;
2
3 a = a + 1;
4 a = a ;
```

Local

- a: 2
- > exports: {}
- > module: Module {id: '.', path...
- > require: f require(path) {\r\...
- __dirname: 'c:\\Users\\esteb\\...

Como podemos ver, en el breakpoint nuestra variable vale “2” y eso lo vemos reflejado en el menú de debug a la izquierda.

3 | Interfaz

Interfaz de debug







Cuando abrimos el menú de debug nos despliega el siguiente menú:

- **Play**
- **Step over**
- **Step into**
- **Step out**
- **Restart**
- **Stop**



Veamos cada parte en más detalle...

Controles

Icono	Función
	Continue: Continúa con la ejecución del código hasta el siguiente breakpoint.
	Step over: Ejecuta la siguiente línea de código. Si es una función, la ejecuta y retorna el resultado.
	Step into: Ejecuta la siguiente línea de código, pero si es una función “entra” a ejecutarla línea por línea.
	Step out: Devuelve el debugger a la línea donde se llamó a la función.
	Restart: Reinicia el debugger.
	Stop: Frena el debug.

DigitalHouse>
Coding School