

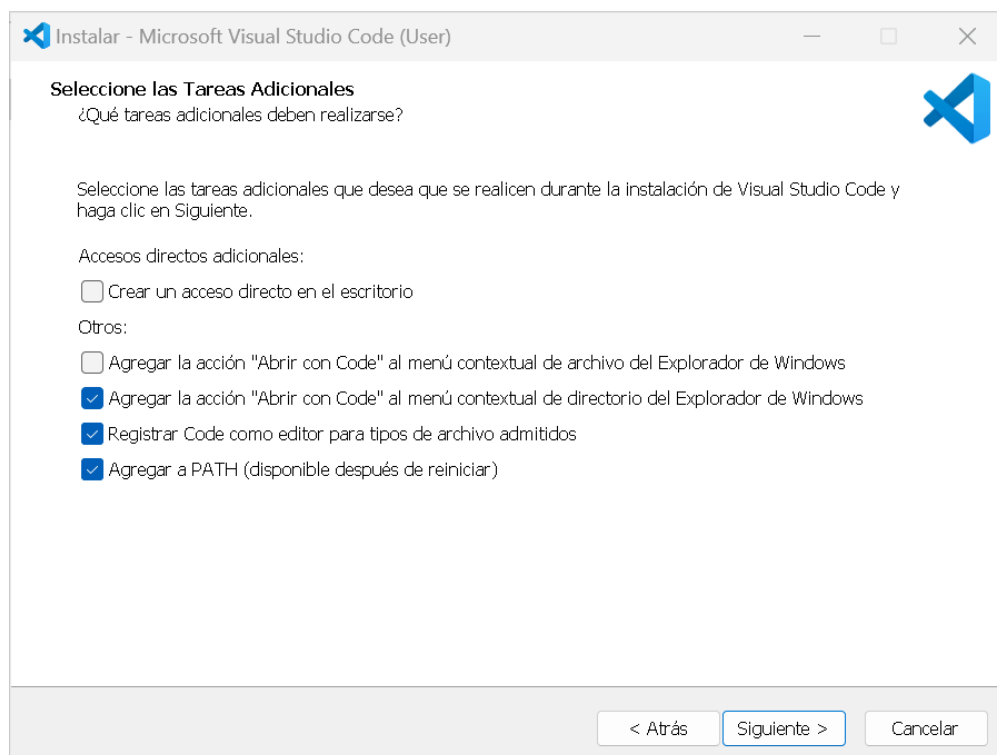
## Primeros pasos JavaScript

### Configurar herramientas desarrollo

Lo primero es instalar un entorno de desarrollo integrado (IDE), en este curso vamos a emplear VS Code por ser una herramienta muy popular que cuenta con multitud de extensiones facilitando las labores de desarrollo.

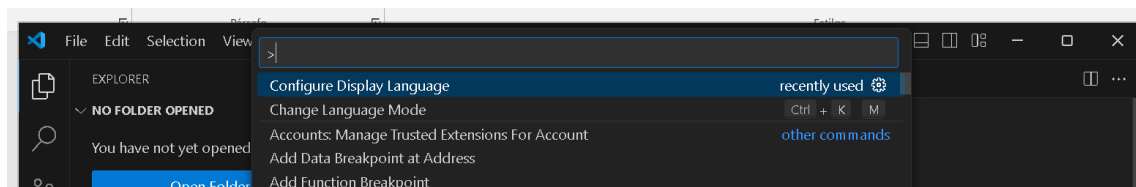
#### Instalar VS Code

Podemos descargar el instalador desde la web <https://code.visualstudio.com/>



Una vez instalado personalizamos el entorno, lo primero cambiar el idioma por defecto a español. Para ello abrimos la barra de búsquedas y escribimos **“Configure Display Language”**, seleccionamos español y reiniciamos el entorno.

**TRUCO:** Un atajo de teclado útil es **“Ctrl + Shift + P”**, que abre la barra de búsquedas.




Seguido instalamos las extensiones más interesantes para desarrollar con JavaScript:

#### ESLint



Es un linter de código JavaScript y JSX, podemos instalarlo manualmente o esperar a que VSCode nos lo ofrezca la primera vez que abramos un fichero con la extensión “.js”.

La página en el market place es

<https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>



### ESLint

Microsoft  [microsoft.com](https://microsoft.com) |  37,690,032 installs | ★★★★★ (237) | Free

Integrates ESLint JavaScript into VS Code.


[Install](#) [Trouble Installing?](#)

### Live Server


Live Server arranca un servidor de desarrollo local con recarga dinámica de las páginas, es decir, puedes ver en el navegador tu proyecto y a medida que modificas el código fuente el navegador se refresca automáticamente.

La página en el market place es

<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>



### Live Server

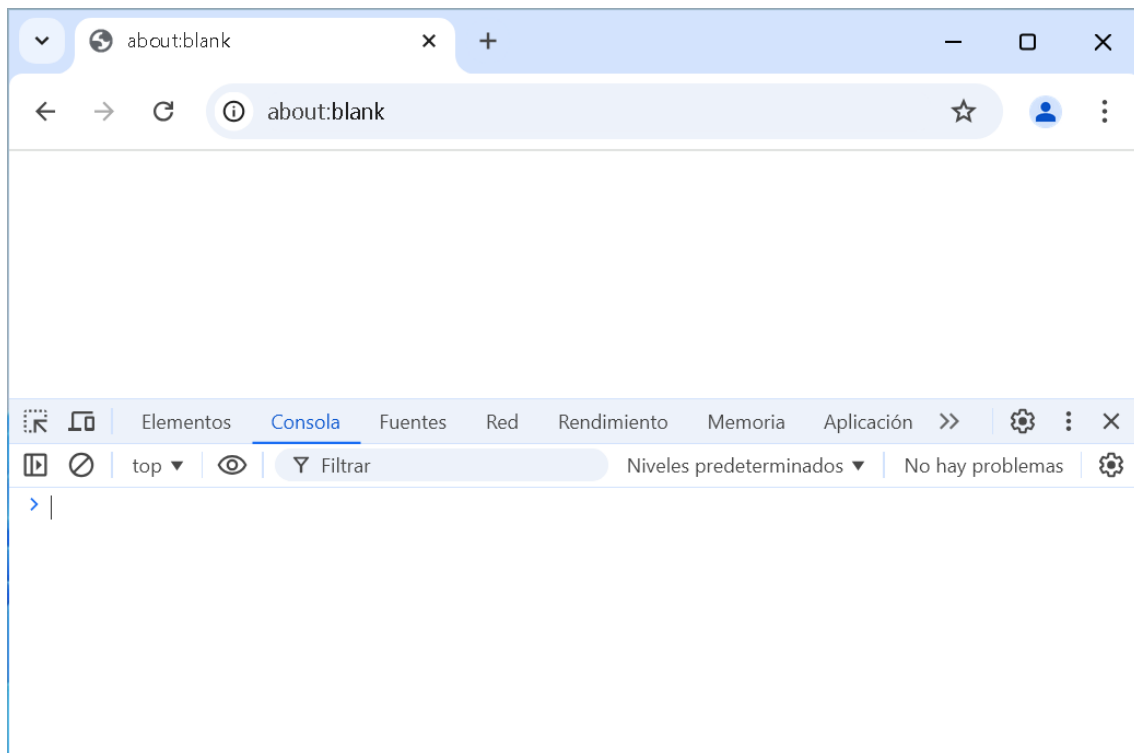
Ritwick Dey |  53,722,716 installs | ★★★★★ (491) | Free

Launch a development local Server with live reload feature for static & dynamic pages

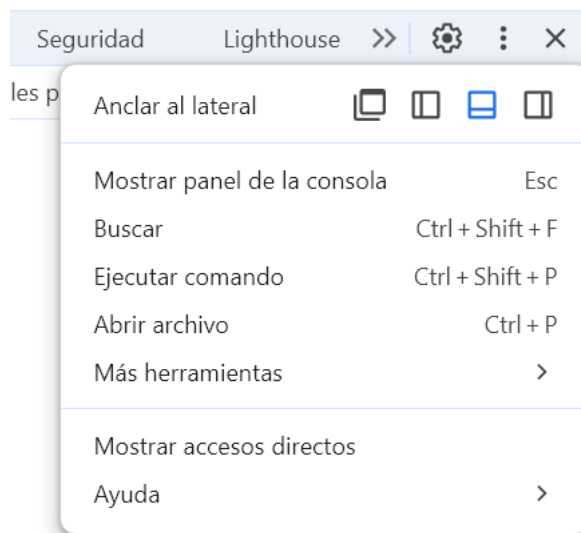
[Install](#) [Trouble Installing?](#)

### Primeros pasos.

Vamos a abrir Chrome (o Chromium) y vamos a navegar a la dirección **"about:blank"**. Verás una página completamente en blanco. Pulsa la tecla de función **F12** que abre las herramientas de desarrollador.

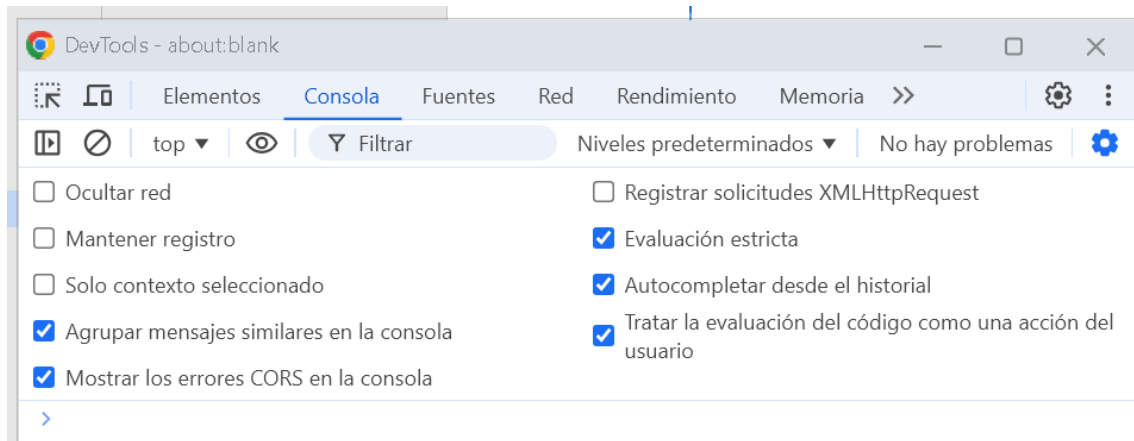


Las herramientas se pueden desacoplar con lo que tendremos mayor área de trabajo. Pulsamos en “desanclar a otra ventana”.



Es importante que marques las siguientes opciones de configuración:

- **“Evaluación estricta”**. JavaScript te avisara cuando cometes errores comunes.
- **“Agrupar mensajes similares en la consola”**. Mantiene la salida en la consola ordenada.



En la pestaña de “Consola”, tenemos el interprete de JavaScript. Desde aquí podemos ejecutar código JavaScript libremente. La flecha izquierda “<” indica una celda de entrada, y al pulsar intro, me devuelve el resultado de evaluar el código en una nueva celda con la flecha derecha “>”.

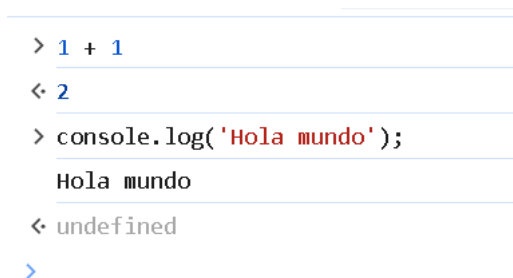
```
< 1 + 1
```

```
> 2
```

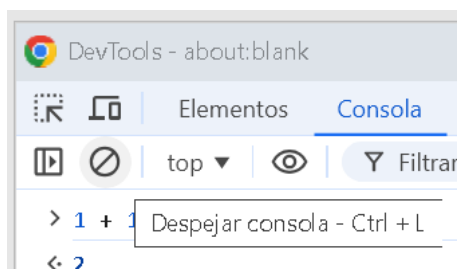
```
> console.log('hola mundo');
```

```
Hola mundo
```

```
< undefined
```

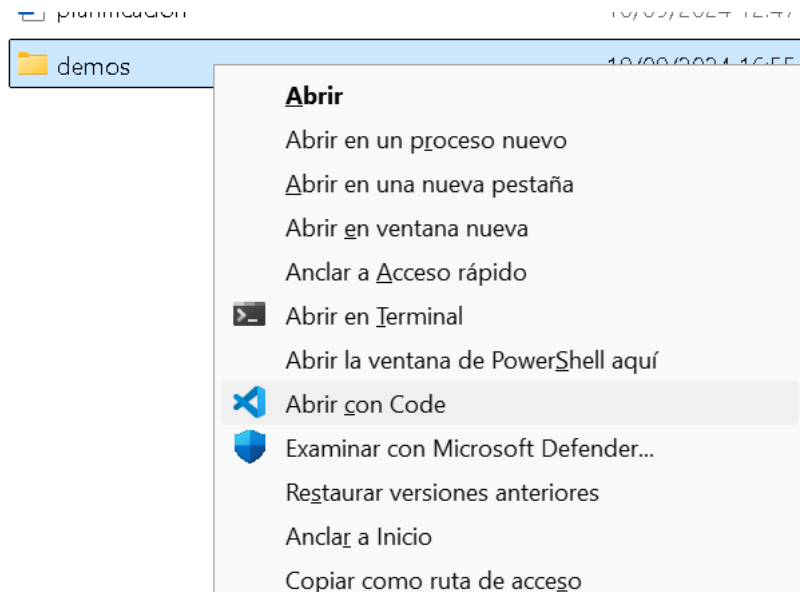


Podemos limpiar la consola pulsando “Ctrl + L”. O pulsando el botón de “Despejar consola”.



Veamos como estructurar un proyecto con VS Code y “Live Server”.

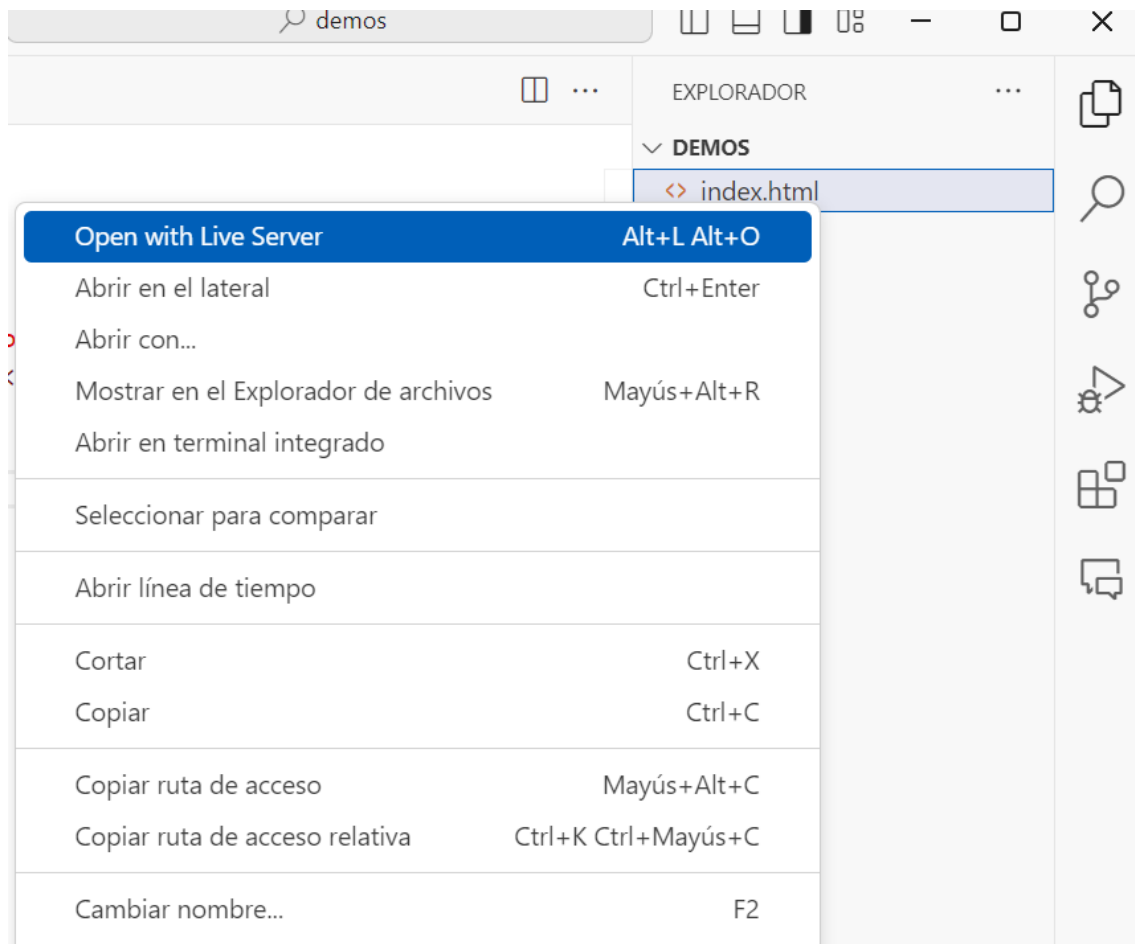
Nos creamos una carpeta que contenga nuestro código, en este caso la hemos llamado “demos”. Y la abrimos con VS Code.



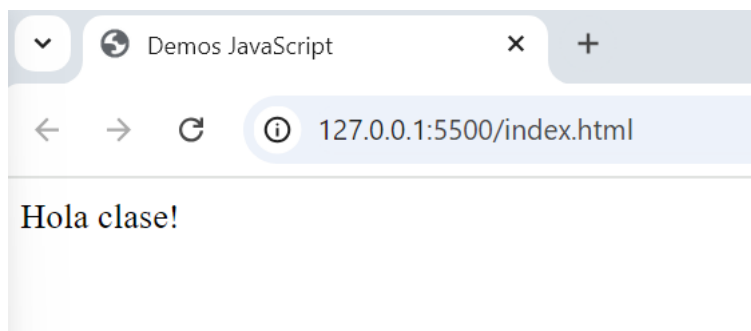
Seguido creamos un fichero “index.html” y le añadimos algo de texto.



Arrancamos el servidor de “Live Server” con el fichero index.html. (También tienes un icono en la barra de estado para arrancarlo y pararlo)



Y si todo va bien se abrirá un navegador con nuestra web. Fíjate que trabajamos en localhost en el puerto 5500, y que el recurso es “index.html”.

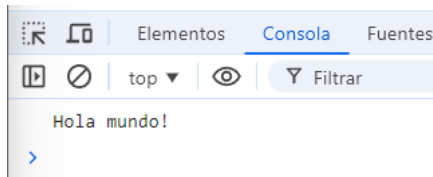


Ahora añadimos nuestro primer código JavaScript, para ello en el `<body>` añadimos la etiqueta `<script>` y escribimos una traza en la consola.

```
<body>
  Hola clase!
  <script>
    console.log('Hola mundo!');
  </script>
</body>
```

Sólo nos falta guardar el archivo en el editor, el atajo “**Ctrl + S**” guarda el fichero actual apréndetelo porque lo vas a usar mucho.

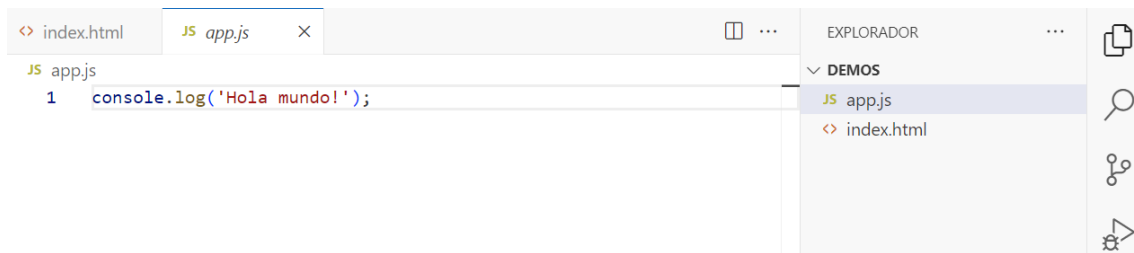
Si compruebas el navegador verás el mensaje en la consola.



**NOTA:** “Live Server” no detecta los cambios cuando la carpeta del proyecto se encuentra en una ubicación de red.

El estándar HTML recomienda separar el contenido (.html), la presentación (.css) y la interacción (.js) en ficheros independientes para facilitar el mantenimiento de la página.

Vamos a llevarnos nuestro script a un fichero independiente, para ello lo primero es crear un nuevo fichero “app.js” en nuestra solución y añadirle el código.



Y en el `<head>` añadimos la referencia al fichero “app.js”.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Demos JavaScript</title>
  <script src="app.js"></script>
</head>
<body>
  Hola clase!
  <!-- <script>
    console.log('Hola mundo!');
  </script> -->
</body>
```

Al guardar todo “Live Server” actualizará nuestro navegador de depuración.