**¿QUE ES BROWSERIFY?**

Browserify es básicamente una herramienta que nos permite gestionar dependencias en forma de módulos del lado del cliente (en el navegador). A grandes rasgos podemos decir que lo que nos permite es crear y requerir módulos tal y como hacemos con Node.js; y digo a grandes rasgos porque no es exactamente así, ya que con browserify no tendremos en el build final varios ficheros javascript, sino que crearemos un bundle con la herramienta y tan solo tendremos un fichero javascript.

primero que nada deberás de tener instalada la herramienta de manera global con npm: 

Una vez hecho esto, podremos ejecutar directamente browserify desde nuestra terminal, es importante que la instalación haya sido global, o en caso contrario no funcionará, puedes hacer la prueba pidiendo la versión del paquete:

Si te ha devuelto un número (en mi caso 8.1.3) es que todo está correcto.

Vamos a crear ahora nuestro primer módulo, que, para no perder las buenas costumbres será el típico “Hola mundo”. Lo llamaremos hello.js:



Ahora creamos otro módulo en el mismo directorio llamado index.js, y hacemos requiere al modulo hello.js:



Todo listo para crear nuestro archivo final, ahora simplemente debemos de ir a nuestro terminal y ejecutar browserify indicando el archivo de entrada, que es el script desde donde tiene que empezar a “seguir” todas las dependencias, y el nombre del fichero final que debe generar, así:

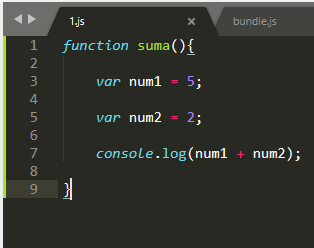


Esto nos generará un script llamado bundle.js (o el nombre que tu quieras) que es el que debemos incluir en nuestro fichero HTML como cualquier otro script. Como podrás ver, el proceso de gestión de módulos es automático, browserify se encarga de seguir todas las dependencias e incluirlas dentro del archivo final, fácil y rápido.

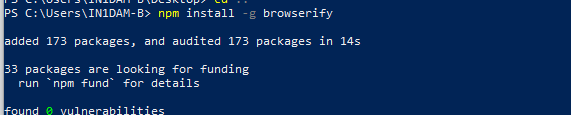
DOCUMENTACIÓN SOBRE BROWSERIFY:

<https://github.com/browserify/browserify#usage>

Creamos un archivo js con una función de prueba

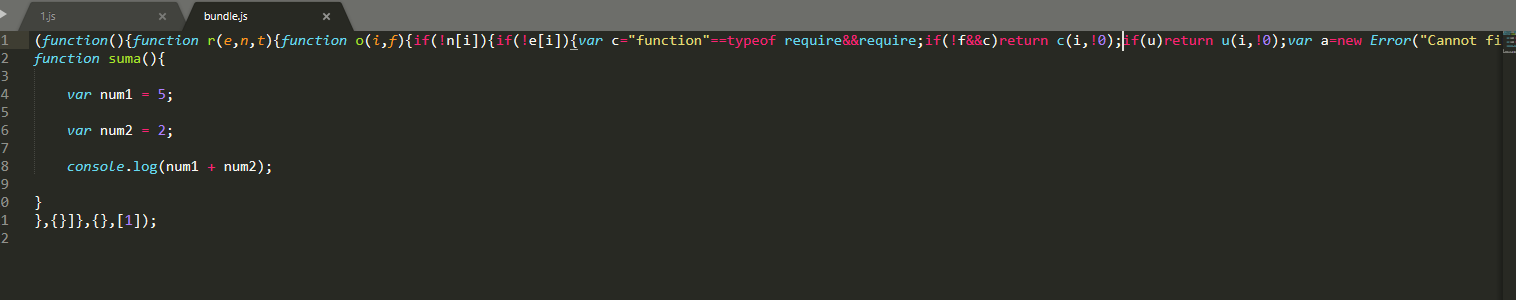


Instalamos el Browserify

****

Generamos el Script llamado “bundle.js” a partir del js que contiene nuestra función

****

****