Depuración de JavaScript.

Práctica 1. Instalación de un Servidor NodeJS para poder subir aplicaciones Angularjs

1.- Desde la página nodejs.org descarga el instalador msi (node-x64.msi) en el caso de Windows.

2.- Sigue el asistente de instalación, en c:\node .
(Tendremos npm para trabajar con paquetes de node)

3.- Abrimos la consola de comandos (prompt) y vemos que se ha llevado acabo bien la instalación.

4.- c:\> node --version → Nos dice la versión instalada

5.- c:\> npm --version → (Node Package Manager) Administrador de paquetes de Node

6.- c:\> npm install -g express-generator → (Instalamos el Servidor Web ExpressJS)
(Con Angular podemos usar Apache, IIS... y otros servidores)

- g para que lo haga global y podamos acceder a express desde cualquier directorio.
  Ahora vamos a crear nuestro primer servidor.

7.- c:\> express miservidor → Es el directorio donde se creará el servidor, con las dependencias que necesita para funcionar.

8.- c:\> cd miservidor → si vas al directorio donde se ha creado el servidor, veras todos los archivos para que funcione. → En public habra que colocar las aplicaciones web.
Ahora para poder iniciar el servidor, lo primero es, cambiarnos al directorio donde se ha creado el servidor. "miservidor"

9.- c:\miservidor> npm install → Instalamos el Administrador de paquetes en nuestro servidor.

10.- c:\miservidor> npm start → Iniciamos el servidor (y empieza a ejecutarse)

11.- Abre el Navegador y escribe en la barra de direcciones: **localhost:3000** → Se abre la página que se crea al arrancarlo "**Express Welcome to Express**"

12.- c:\miservidor> cd public → Me cambio al directorio public dentro de miservidor. Y ahi es donde copiaria los archivos que forman nuestra aplicación:
"**index.html, styles.css, scripts.js**"

13.- c:\miservidor\public> → Copia aquí los archivos de la aplicación "angular7" hml, js y css.
Aquí dentro también podría poner el archivo angular.min.js

14.- Abre el Navegador y escribe: **localhost:3000/index.html**
localhost:3000/angular7.html
→ Y como ves se ejecuta la aplicación desde el servidor.

Práctica 2. Utilización de un Automatizador (Ejecutor) de Tareas de JavaScript (Grunt), que nos permite depurar el código, reducir el código CSS y JavaScript, mezclar archivos.
Es cómo instalar un módulo de nodejs para ello usamos npm.

Instalamos el "comand line interface" = cli (interfaz de linea de comandos del propio grunt)

1.- Desde la consola de Comandos con derechos de administrador: C:\> npm install -g grunt-cli
El hecho de instalar grunt-cli no instala automáticamente grunt y sus dependencias, debido a que podemos tener instaladas más de una versión de grunt.

2.- Para instalar grunt creamos un directorio c:\> md miproyecto

3.- Cambiate al directorio y creamos un archivo: c:\miproyecto> package.json (ojo extensión)

4.- Escribe dentro del archivo, las características de la aplicación, nombre, versión, y las dependencias que se necesitan instalar: (copia el código de la pagina de grunt)

```
{
        "name":"miproyecto",
        "version":"0.1.0",
        "devDependencies": {
                "grunt":"~0.4.5",
                "grunt-contrib-jshint": "~0.10.0",
                "grunt-contrib-concat": "~0.4.0",
                "grunt-contrib-uglify": "~0.5.0",
                "grunt-shell": "~0.7.0"
        }
}
```

Nombre del proyecto, numero de la versión.
La lista de dependencias necesarias para la aplicación.
En la pagina de grunt estan los plugins y las ultimas versiones.

5.- c:\miproyecto> npm install grunt --save-dev → Instalamos grunt en el directorio del proyecto

6.- c:\miproyecto> grunt --version → Para comprobar que se ha instalado correctamente tiene que aparecer grunt, y grunt-cli

7.- c:\miproyecto> dir → Dentro de miproyecto se han creado: node_modules y package.son dentro del directorio node_modules, estan todas las dependencias que necesitamos.

8.- Ahora podemos crear una tarea grunt y para ello necesitamos un archivo JavaScript, ya que en principio vamos a depurar codigo JavaScript.

Práctica 3. Depurar el Código con Grunt.

Para ello, usamos el archivo "angular7.js" y vamos a crear un archivo "**gruntfile.js**" que es donde hacemos una lista de las tareas que necesitamos que el grunt lleve a cabo.
Compruebo que el código "angular7.js" está depurado de errores y para eso utilizamos "**jshint**"
Instalamos el inicializador de grunt

1.- c:\miproyecto> npm install -g grunt-init (ojo hay que estar como administrador)
npm install grunt-contrib-jshint --save-dev

2.- Creamos el archivo "**gruntfile.js**" (dentro de ese archivo especificamos el archivo que queremos depurar scripts.js) en la carpeta donde se encuentre "package.json" y escribimos el código …. (ya lo tienes en un archivo)
Ahi, en la carpeta de mi proyecto tambien ponemos el archivo .js donde queremos detectar errores.

3.- c:\miproyecto> grunt
→ Y nos indica los posibles errores (nos indica por ejemplo que faltan ";" (semicolon)
→ Así, que modificamos en nuestro archivo "angular7.js" los errores
→ Guardamos el archivo y volvemos a ejecutar c:\miproyecto> grunt → No hay errores

Práctica 4. Mezclar y Reducir archivos .js con Grunt

Supongamos que tenemos tres archivos: "angular7.js", "script1.js", "script2.js"
Incluiremos dos tareas:

- Una para mezclar los tres archivos.
- Otra para reducir ó minimizar su tamaño.

1.- Editamos el archivo "gruntfile.js" y lo modificamos:
Añadimos el código de las tareas nuevas ….concat: uglify:

2.- c:\miproyecto> grunt → Entonces grunt ejecutará las tareas que tiene.

3.- c:\miproyecto> dir → Vemos que está creado el fichero "unidos.js"

4.- Abre en el editor el archivo "unidos.js" → Verás que dentro está el código de los tres archivos

5.- c:\miproyecto> cd build → Cambiate a la carpeta "build" → Y se ha creado "unidos.min.js"
→ El código está minimizado y reducido.