Se adjuntan archivos referentes a la actividad 1 en un archivo .zip y respuestas aqui abajo.

El .zip contiene los ficheros javascript, el html, el md, respuestas en txt y capturas de pantalla.

Para correrlo, basta con acomodar los archivos en una carpeta individual, correr el live server, abrir consola de navegador (preferente chrome porque en mozilla no se me quisieron imprimir) y observar el resultados.

Los cambios se subieron a mi fork en el siguiente link: https://github.com/engmmcasas/Javascript-Course/tree/main/activity/1_session

##

##

##

##
index.js:5 ##
index.js:6 Ejercicio 1
aboutme.js:11 Hola, me llamo Manuel Martinez Casas y me dedico a Ingenieria en Inteligencia Artificial.
Estoy cursando este Máster porque me gustaría trabajar en Desarrollo Web
index.js:10 ##
index.js:11 ##
index.js:13 Ejercicio 2
index.js:14 Se ejecuta primero el contenido de aboutme.js
index.js:19 ##
index.js:20 ##
index.js:22 Ejercicio 3
index.js:48 Monitor de red

Es la herramienta del navegador que se encarga de detallar las solicitudes hechas, su evolución temporal, y sus caracteristicas.

Contempla la lista de solicitudes donde se muestran el codigo regresado, el tipo de metodo , el nombre del archivo, la ruta de donde se pidio, el tipo de contenido y tamaño. Igualmente, cada uno tiene una barra de tiempos apilados que muestras la evolucion de la solicitud.

Al ver el detalle de la alguna solicitu se podra ver sus encabezados, cookies, parametros, respuestas y la evolucion temporal en detalle.

Memoria

La herramienta de memoria permite tomar una captura del tamaño de cada uno de las partes de la pagina web, categorizarlos como en strings, scripts, otros y objetos (y en la version que tengo en uno mas que es domNode), y mostrarlos ya sea como mapa de arboles, vista agregador o dominadores, con el ultimo fin de poder saber que secciones de la pagina toman mas o menos memoria y tomar acciones en caso de que sea requerido.

Performance

La herramienta de performance tiene la tarea de evaluar la eficiencia de la pagina y los elementos que ocurren en esta. Se tiene tres herramientas internas llamadas Frame Rate, Waterfall, Call Tree y Frame Chart.

El frame rate es ampliamente utilizado en videojuegos y videos por ser la tasa de imagenes producidas en un intervalo de tiempo y actualmente se utiliza mucho para evaluar la responsividad de una pagina web. De acuerdo a la documentacion, una buena meta es 60 fps pero si otro proceso estorba para cumplir el objetivo, se puede ver facilmente en esta vista y usar las otras herramientas para tener un mejor diagnostico para solucionarlo.

El Waterfall evidencia los procesos hechos por el navegador como repaints, ejecuciones de javascript, etc, los distingue en diferentes categorias, da detalles de estos y mide el tiempo que tomo.

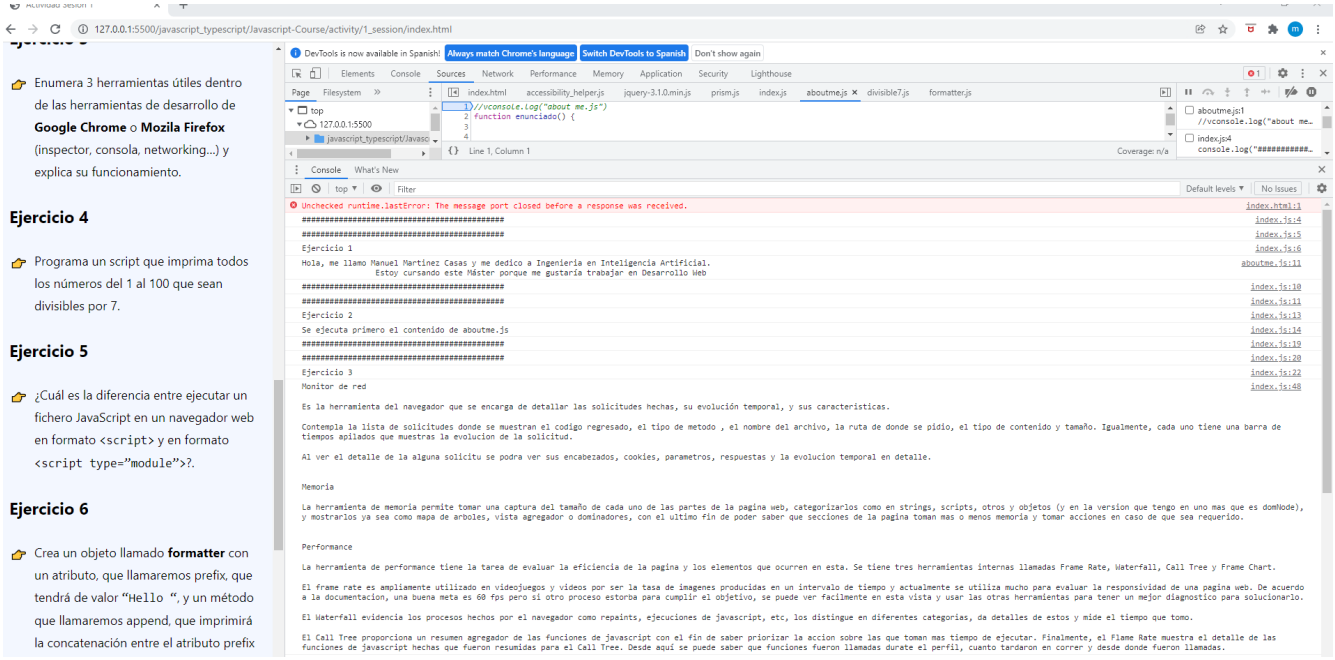
El Call Tree proporciona un resumen agregador de las funciones de javascript con el fin de saber priorizar la accion sobre las que toman mas tiempo de ejecutar. Finalmente, el Flame Rate muestra el detalle de las funciones de javascript hechas que fueron resumidas para el Call Tree. Desde aquí se puede saber que funciones fueron llamadas durate el perfil, cuanto tardaron en correr y desde donde fueron llamadas.

```
index.js:51 ##############################################
index.js:52 ##############################################
index.js:53 Ejercicio 4
divisible7.js:7 7
divisible7.js:7 14
divisible7.js:7 21
divisible7.js:7 28
divisible7.js:7 35
divisible7.js:7 42
divisible7.js:7 49
divisible7.js:7 56
divisible7.js:7 63
divisible7.js:7 70
divisible7.js:7 77
divisible7.js:7 84
divisible7.js:7 91
divisible7.js:7 98
index.js:58 ##############################################
```

index.js:59 ##
index.js:60 Ejercicio 5
index.js:70 Las principales diferencias son:

1. Usando el tipo "module" se especifica que el archivo javascript utiliza modulos y puede importar estos desde otros ficheros.
2. El modo estricto esta activado usaod el tipo "module", mientras que cuando no esta se tiene que especificar.
   Hay ciertos modos parecidos a otros lenguajges en modo estricto, como el error de una variable no declarada previamente.
3. Cada ficher en modo "module" tiene su pripio "scope", mientras que sin este, las variable se hacen globales.

index.js:72 ##
index.js:73 ##
index.js:76 Ejercicio 6
formatter.js:6 Hello Master Full Stack Development
formatter.js:12 hola mundo



- Enumera 3 herramientas útiles dentro de las herramientas de desarrollo de **Google Chrome** o **Mozila Firefox** (inspector, consola, networking...) y explica su funcionamiento.

Ejercicio 4

- Programa un script que imprima todos los números del 1 al 100 que sean divisibles por 7.

Ejercicio 5

- ¿Cuál es la diferencia entre ejecutar un fichero JavaScript en un navegador web en formato `<script>` y en formato `<script type="module">`?.

Ejercicio 6

- Crea un objeto llamado **formatter** con un atributo, que llamaremos prefix, que tendrá de valor "`Hello `", y un método que llamaremos append, que imprimirá la concatenación entre el atributo prefix