Badajoz, 13 de junio de 2022

Summer Quiz

Práctica final del módulo de JavaScript avanzado del Bootcamp de FSWD

Por: **Alejandro Rodríguez** y **Javier Guerra**

Repositorio en GitHub: https://github.com/JavGuerra/javascript-fswd

Enunciado:

https://github.com/TheBridge-FullStackDeveloper/fullstack_INDRA_PT_mar22/blob/main/teoria/fundamentos-javascript/clase10.md

Planificación: Fases de la aplicación

Fase 1: Pantalla de bienvenida

Duración: 3 jornadas, días 1, 2 y 3.
Pasos:

1. Las fases 2 y 3 deben estar ocultas en el HTML (vía CSS).
2. Debe haber un mensaje de información y bienvenida.
3. Si hay puntuaciones guardadas en localStorage, se muestran en una tabla.
4. Con los datos de la tabla se puede mostrar una gráfica.
5. Debe haber un botón para iniciar el cuestionario de la fase 2.

Fase 2: Cuestionario

Duración: 5 jornadas, días 6, 7, 8, 9 y 10.
Pasos:

1. Deshabilitar fase 1, y habilitar fase 2.
2. Realizar consulta a la API para obtener 10 preguntas y guardar en Array.
3. Ir mostrando cada una de las preguntas con sus opciones. Pasos:
   3.1 Leer la pregunta del Array de preguntas.
   3.2 Mostrar el enunciado de la pregunta.
   3.3 Mostrar las cuatro opciones en la misma pantalla.
   3.4 Mostrar un botón de enviar.
   3.5 Esperar a que el usuario apriete el botón. Después comprobar:
      5.1 Si acertó la opción, contabilizar el acierto en una variable.
      5.2 Si no la acierta, no hay que hacer nada, y se pasa a la siguiente.

5.3 Si no contestó la pregunta: error, ya que todas las preguntas son obligatorias.
Opción: no habilitar el botón enviar hasta contestar.

4 Cuando se termine el cuestionario (10 preguntas), debe saltar automáticamente a la fase 3.

Fase 3: Finalización del cuestionario

Duración: 1 jornada, día 2.
Pasos:

1 Deshabilitar fase 2 y habilitar fase 3.
2 Presentación del resultado del cuestionario: aciertos sobre 10.
3 Guardar el resultado en localStorage.
4 Debe haber un botón para volver a jugar.

Fase de preparación y revisión

Duración: 3 jornadas, días 31, 4 y 11.
Pasos:

1. planificación del trabajo.
2. Test de uso y presentación.

Calendario

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
	31	1	2	3	4
	Enunciado y planificación.	Maquetación básica de las tres fases.	Uso de localStorage. (Fases 1 y 3)	Tabla de resultados.	Revisión del trabajo de las fases 1 y 3.
6	7	8	9	10	11
Consultar la API y obtener las preguntas.	Mostrar 10 preguntas consecutivas.	Validaciones y diseño CSS avanzado.	Añadir preguntas propias JSON.	Otros posibles extras. (gráfica)	Revisión final y preparar presentación.
13					
Entrega					

Características destacables de este ejercicio

Operativa de la aplicación

- Aplicación de una sola página (SPA).
- Integración de 25 preguntas propias en un JSON. Usa una en cada partida.
- Hace uso de localStorage pata guardar y leer las puntuaciones.
- Gráficas dinámicas en SVG de elaboración propia y uso de DOM para su gestión.
  - Gráfica de líneas para mostrar estadísticas. Manejo del id ‘theline’.
  - Contador visual para mostrar el resultado final del quiz. Manejo del id ‘hand’.
- Animaciones CSS de gráfica de líneas y personajes.
- Carga de las gráficas SVG al inicio de la app con fetch para hacer uso de los id que incorporan y poder así hacer cambios dinámicos. Con <img> esto no funcionaría.
- Los posibles errores se muestran a través de una ventana modal <dialog>.
- Usa JavaScript puro (Vanilla JS) sin frameworks.
- Recopila librería de funciones propia en el fichero js/functions.js
- Repositorio compartido en Github y web alojada en Github pages.

Usabilidad

- La web implementa los criterios del estándar XHTML (revisado con Nu Html Checker).
- El diseño es adaptable (responsive) según el dispositivo.
- Es accesible (revisado con el complemento WAVE):
  - Hace uso de etiquetas WAI-ARIA para describir eventos interactivos.
- Tiene temática coherente y un cuidado aspecto visual trabajado con CSS.
- Interfaz en inglés acorde al contenido de la API.
- Emplea un '*spin*' asíncrono para informar al usuario de que se está gestionado la consultas.
- Los botones no se activan hasta que terminan los procesos asíncronos: cargas, consultas...
- La selección de las opciones de cada pregunta se hace a través de botones de radio cuyo aspecto modificado con CSS simula el aspecto de botones de selección.
- Muestra barra de puntuaciones medias y barra de progreso del questionario.
- Implementa el protocolo Open Graph (*head*) para la correcta inserción de la web en RRSS.
- Es compatible con *Progessive Web Aplication* (PWA) a través de fichero manifest.json, por lo que la página puede ser instalada en dispositivos móviles como una web app.
- Se dispone de un código QR para acceso rápido a la web a través de dispositivos móviles.
- Ha sido probada con los navegadores web Firefox y Chrome.
- Extra: La consola del navegador muestra ayuda con las respuestas.

Acceder a la página web

Página web del ejercicio:
https://javguerra.github.io/javascript-fswd/

Licencias

Se hace uso de la API de Open Trivia DB: https://opentdb.com/