FRONT-END

EXAMEN 1

Víctor Carazo Contreras

1. Definiciones breves (responde en una línea cada una)

a) Es una técnica que permite encapsular el estilo y la estructura de un componente para que no afecte ni sea afectado por el DOM principal.

b) Son reglas en CSS que aplican estilos diferentes según el tamaño o las características del dispositivo o pantalla.

c) Devuelve el número de milisegundos transcurridos desde el 1 de enero de 1970 (época Unix).

d) Indica que el script se cargará en segundo plano y se ejecutará solo cuando el documento HTML haya terminado de analizarse.

2. Verdadero/Falso y justifica (si es falso, corrige la idea en una frase)

a) Falso. Usar innerHTML no es seguro porque puede provocar ataques XSS; se debe usar textContent o sanitizar el contenido.

b) Verdadero. Genera números enteros entre 0 y 9 incluidos.

c) Verdadero. Un selector con id tiene mayor especificidad que uno con varias clases.

d) Falso. Las funciones flecha no tienen su propio this, sino que heredan el del contexto donde fueron definidas.

3. Selección DOM moderna. Nombra dos métodos modernos para seleccionar elementos del DOM y menciona una ventaja de usar querySelector frente a getElementById.

Dos métodos modernos: querySelector() y querySelectorAll().

La ventaja de querySelector() es que permite usar cualquier selector CSS (clases, ids, etiquetas o combinaciones), mientras que getElementById() solo busca por id.

4. **Eventos. Explica la diferencia entre event.preventDefault() y event.stopPropagation(), y da un ejemplo práctico de cuándo usarías cada uno.**

   event.preventDefault() evita que ocurra la acción por defecto del evento.

   event.stopPropagation() detiene la propagación del evento hacia los elementos padre.

Ejemplo:

- **preventDefault():** en un formulario, para validar datos antes de enviarlo.
- **stopPropagation():** en un botón dentro de un contenedor con otro evento, para que el clic no active también el evento del contenedor.

5. **Buenas prácticas de eventos. Menciona dos ventajas de usar event delegation frente a asignar manejadores a elementos individuales.**

   Dos ventajas que se me ocurren son:

   - **Mejor rendimiento**, porque se usa un solo listener en lugar de muchos, y así se reduce el consumo de memoria.
   - **Mayor flexibilidad**, ya que, los nuevos elementos añadidos dinámicamente también responden al evento sin necesidad de reasignarlo.

6. **Manipulación del DOM. ¿Cuál es la diferencia entre element.innerHTML y element.outerHTML? ¿Cuándo es preferible usar insertAdjacentHTML?**

   innerHTML modifica o devuelve el contenido dentro de un elemento, mientras que outerHTML incluye también al propio elemento.

   insertAdjacentHTML se usa preferiblemente cuando quieres insertar HTML sin reemplazar el contenido existente, eligiendo la posición exacta donde añadirlo.

7. **CSS: selectores. Escribe qué selecciona cada uno:**

   a) ul >li

   b) .btn:disabled

   c) nav ul li:first-child a

   a) Selecciona los elementos <li> que son hijos directos de un <ul>.
   b) Selecciona los elementos con clase .btn que están deshabilitados.
   c) Selecciona los enlaces <a> dentro del primer <li> de cada lista <ul> que esté dentro de un <nav>.

8. **CSS: modelo de cajas. Explica la diferencia entre box-sizing: content-box y box-sizing: border-box. ¿Cuál se considera mejor práctica hoy día y por qué?**

   En content-box, el ancho y alto se aplican solo al contenido, sin contar bordes ni padding.
   En border-box, el tamaño incluye contenido, padding y borde.
   Hoy se prefiere border-box porque facilita el diseño y evita cálculos al ajustar tamaños.

9. **Responsive. ¿Qué son los breakpoints en diseño responsive? Explica cómo se relacionan con el concepto de mobile first.**

   Los breakpoints son puntos definidos en el ancho de la pantalla donde el diseño cambia para adaptarse mejor al dispositivo.
   En el enfoque mobile first, se diseñan primero los estilos para pantallas pequeñas y se usan breakpoints con min-width para ampliar el diseño a pantallas mayores.

10. **Bootstrap 5 (rejilla). Responde breve:**

    a) ¿Qué clase se usa para definir una fila en Bootstrap?
    b) ¿Qué indica el prefijo en .col-md-4?
    c) Escribe dos clases de utilidades para espaciado (margin/padding)

    a) .row
    b) Que la columna ocupa 4 de 12 espacios en pantallas medianas (≥768px).
    c) m-3 (margin) y p-2 (padding)

11. **JavaScript: tipos y comparaciones. ¿Por qué null == undefined devuelve true pero null === undefined devuelve false? Explica la regla de coerción.**

    null == undefined devuelve **true** porque el operador == realiza coerción de tipos y los considera equivalentes cuando solo uno de ellos es null o undefined.
    null === undefined devuelve **false** porque === compara tipo y valor, y null y undefined son tipos distintos.

12. **Funciones flecha. Menciona dos limitaciones de las funciones flecha respecto a las funciones tradicionales (function).**

    - No tienen su propio this, heredan el contexto donde se definen.
    - No pueden usarse como constructores ni tener métodos como arguments.

13. **Parámetros y retorno. Explica brevemente el spread operator y destructuring assignment. Pon un ejemplo de una línea para cada caso.**

    Spread operator: permite expandir elementos de un array u objeto.

    Destructuring assignment: permite extraer valores de arrays u objetos en variables.

14. **Manejo de errores. ¿Cuál es la diferencia entre Error y TypeError? Menciona una buena práctica al usar bloques try-catch.**

    Error es el tipo general de error en JavaScript, mientras que **TypeError** ocurre cuando se realiza una operación sobre un tipo de dato inapropiado.

    Buena práctica: capturar solo los errores esperados y registrarlos o manejarlos sin silenciar fallos inesperados.

15. **Fecha y tiempo en JS. ¿Qué método usarías para formatear una fecha en un formato legible para el usuario? Nombra una librería externa común para manejo avanzado de fechas.**

    Para formatear una fecha en un formato legible se puede usar toLocaleDateString() o toLocaleString().

    Una librería externa común es date-fns o Moment.js.

16. **Aleatoriedad. ¿Por qué Math.random() no es adecuado para aplicaciones criptográficas? Nombra una alternativa nativa más segura.**

    Math.random() no es seguro porque los númerosque genera son pseudoaleatorios y pueden predecirse.

    Una alternativa nativa más segura es crypto.getRandomValues().

17. **Módulos. Explica la diferencia entre export default y export con nombre. Añade un ejemplo de importación para cada caso.**

    export default permite exportar un único valor por archivo y se importa sin llaves.

    - Ejemplo:

      ```
      export default function prueba() {}
      import prueba from './modulo.js';
      ```

    export con nombre permite exportar varios elementos y se importan con llaves.

    - Ejemplo:

      ```
      export const PI = 3.14;
      import { PI } from './modulo.js';
      ```

18. **TypeScript (visión). ¿Qué son los tipos genéricos en TypeScript y cómo contribuyen a la reutilización de código?**

    Los tipos genéricos permiten crear funciones o clases que trabajan con diferentes tipos de datos sin perder seguridad de tipo.

    Ayudan a la reutilización porque un mismo código puede adaptarse a varios tipos sin duplicarse.

19. **Accesibilidad y SEO en formularios. Menciona dos atributos ARIA importantes para mejorar la accesibilidad en formularios y explica brevemente su propósito.**

    aria-label: proporciona una descripción accesible cuando no hay una etiqueta visible.

    aria-required: indica que un campo es obligatorio para completar el formulario.

20. Criterios de calidad de una entrega. Nombra dos métricas de Core Web Vitals y explica cómo afectan cada una a la experiencia de usuario.

LCP (Largest Contentful Paint): mide cuánto tarda en mostrarse el contenido principal. Si es lento, el usuario percibe la página como pesada.

CLS (Cumulative Layout Shift): mide la estabilidad visual; un valor alto provoca saltos molestos en la interfaz mientras carga.