PRUEBA PRÁCTICA FINAL UNIDAD FORMATIVA - 1ª CONVOCATORIA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Código oficial Lanbide | **24201266** | | Fecha prueba: | Antepenúltimo día de la UF | |
| Nombre acción formativa: | **IFCD0110 – CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB** | | | | |
| Código Módulo formativa: | **MF0951\_2** | Código Unidad formativa: | | | **UF1305** |
| Nombre alumno/a: | **Liz Katherine Cebrecos Gayoso** | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **VARIABLES Y CRITERIOS A EVALUAR** | | Puntuación  Máxima | **OBTENIDO** |
| PROCESO | C1 | 4 |  |
| HABILIDAD | C1 | 4 |  |
| FINALIZACIÓN | C1 | 2 |  |
| **Puntuación MÍNIMA para superar la prueba** | | **5** | |
| **Puntuación MÁXIMA de la prueba** | | **10** | |
| **Puntuación GLOBAL OBTENIDA** | |  | |

## ENUNCIADO DEL EJERCICIO

Se requiere desarrollar un slider de imágenes interactivo utilizando JavaScript, HTML y CSS. El componente debe gestionar la visualización de imágenes y sus títulos correspondientes mediante el uso de arrays como estructura de datos principal.

## ESPECIFICACIONES TÉCNICAS

### Datos de entrada

El ejercicio debe implementarse utilizando los siguientes arrays predefinidos:

javascript

const imagenes = [   
 '<https://picsum.photos/800/400?random=1>',  
 '<https://picsum.photos/800/400?random=2>',  
 '<https://picsum.photos/800/400?random=3>',  
 '<https://picsum.photos/800/400?random=4>',  
 '<https://picsum.photos/800/400?random=5>'  
];

const titulos = [  
 'Paisaje Natural',  
 'Arquitectura Moderna',  
 'Vida Urbana',  
 'Naturaleza Salvaje',  
 'Arte Contemporáneo'

];

## DESARROLLO DEL EJERCICIO

### Fase 1: Estructura HTML (20 puntos)

Crear la estructura HTML necesaria que incluya:

* Un contenedor principal para el slider
* Un elemento para la visualización de la imagen actual
* Un elemento para mostrar el título correspondiente
* Botones de navegación para avanzar y retroceder
* Indicadores de posición opcionales

### Fase 2: Diseño CSS (20 puntos)

Implementar los estilos necesarios para:

* Establecer dimensiones fijas para el contenedor del slider
* Configurar la imagen para que se ajuste correctamente al contenedor
* Aplicar estilos apropiados a los botones de navegación
* Implementar transiciones suaves entre cambios de imagen
* Diseñar los indicadores de posición si se incluyen

### Fase 3: Lógica JavaScript (40 puntos)

Desarrollar la funcionalidad mediante:

* Declaración de variable para el control del índice actual
* Función para mostrar la imagen y título actuales
* Implementación de funciones de navegación (anterior/siguiente)
* Asignación de event listeners a los botones de navegación
* Gestión de casos límite en la navegación cíclica (Deshabilitar el botón siguiente o que pase a la primera imágen)

### Fase 4: Funcionalidad complementaria (20 puntos)

Implementar características adicionales:

* Sistema de reproducción automática
* Controles de pausa y reanudación
* Navegación directa mediante indicadores

## REQUISITOS OBLIGATORIOS

1. Visualización correcta de imagen y título actuales
2. Navegación manual mediante botones anterior y siguiente
3. Navegación cíclica completa
4. Transiciones suaves entre elementos
5. Código estructurado y documentado

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Presentación</title>

    <style>

    \*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body{

    background-color: #f0f0f0;

    /\* text-align: center; \*/

}

#cabecera {

      background: #0848d3;

      color: #fff;

      padding: 20px;

      display: flex;

      align-items: center;

    }

    .slider-container {

      position: relative;

      width: 250px;

      height: 300px;

      margin: 40px auto;

    }

#slider {

      width: 100%;

      height: 100%;

      object-fit: cover;

      transition: opacity 0.5s ease-in-out;

    }

    #deslizante { /\* Coloca sombra atrás del texto de la imágen \*/

      position: absolute;

      bottom: 20px;

      left: 0;

      right: 0;

      background: rgba(0,0,0,0.5);

      color: #fff;

      padding: 10px;

      font-size: 1.5em;

      text-align: center;

    }

    .botones {

      margin-top: 20px;

    }

    .botones button {

      padding: 10px 20px;

      margin: 0 10px;

      font-size: 1em;

      cursor: pointer;

    }

    .botones button:hover {

        background: greenyellow

    }

    .indicadores {

      margin-top: 20px; /\* margen de los indicadores \*/

    }

    .indicadores span {

      display: inline-block;

      width: 15px;

      height: 15px;

      margin: 0 5px;

      background: #eca7e7;

      border-radius: 50%;

      cursor: pointer;

    }

    .indicadores .activo {

      background: #3293e2;

    }

    </style>

</head>

<body>

    <header id="cabecera">

        <img src="imagenes/marca.jpg" width="100" title="Logotipo">

        <h1 class="titcab">Visiones enimágticas</h1>

    </header>

    <div class="slider-container">

        <img id="slider" src="" alt="imagen del slider">

        <h2 id="deslizante"></h2>

        </div>

    <div class="controles">

        <div class="botones">

            <button onclick="siguiente()">Siguiente</button>

            <button onclick="anterior()">Anterior</button>

            <button onclick="pausar()">Pausar</button>

            <button onclick="reanudar()">Reanudar</button>

        </div>

        <div class="indicadores" id="indicadores"></div>

        </div>

    <script>

        const imagenes=['Tcn.jpg','universo.jpg','mujerindigena.png','Ind.jpg','Selvabaja.png','Pm.jpg']; /\* imágenes del slider \*/

        const texto=['Tucán','Universo','Shipiba','Sabio','Medicina natural','Puma']; /\* Texto de las imágenes \*/

        let indiceActual = 0; /\* Índice de la actual imágen \*/

        const sliderImage = document.getElementById('slider');

        const sliderText = document.getElementById('deslizante');

        const indicadoresContainer = document.getElementById('indicadores');

        /\* Botones: \*/

        const btnAnterior = document.getElementById('btnAnterior');

        const btnSiguiente = document.getElementById('btnSiguiente');

        const btnPlay = document.getElementById('btnreanudar');

        const btnPause = document.getElementById('btnpausar');

        function mostrarImagen() {

            sliderImage.style.opacity = 0; /\* transición que inicia el desvanecimiento \*/

            setTimeout(() => {

                const imagenSrc = `imagenes/${imagenes[indiceActual]}`;

                const textoActual = texto[indiceActual];

                /\* Actualiza el contenido del slider \*/

                sliderImage.src = imagenSrc;

                sliderImage.alt = textoActual;

                sliderText.textContent = textoActual;

                sliderImage.style.opacity = 1; /\* Al finalizar la presentación hace visible la nueva imágen \*/

                actualizarIndicadores();

            }, 300); // Este tiempo debe coincidir con la transición en CSS

        }

        function iniciarAuto() { /\* Imágenes pasan en automático \*/

            intervalo = setInterval(siguiente, 300);

        }

        function siguiente() {

            indiceActual = (indiceActual + 1) % imagenes.length; /\* El índice comienza en cero al finalizar \*/

            mostrarImagen();

        }

        function anterior() { /\* Se mueve el slider a la siguiente imágen \*/

            indiceActual = (indiceActual - 1 + imagenes.length) % imagenes.length; /\* Va al final y retrocede \*/

            mostrarImagen();

        }

        function pausar() {

            clearInterval(intervalo);

        }

        function reanudar(){

            iniciarAuto();

        }

        function actualizarIndicadores() { /\* Resalta los indicadores que se muestra en ese momento \*/

            const dots = indicadoresContainer.children;

            for (let i = 0; i < dots.length; i++) {

                dots[i].classList.remove('activo');

            }

            dots[indiceActual].classList.add('activo');

        }

        imagenes.forEach((\_, index) => { /\* Se ejecuta al cargar la pantalla \*/

            const dot = document.createElement('span');

            dot.addEventListener('click', () => {

                indiceActual = index;

                mostrarImagen();

            });

            indicadoresContainer.appendChild(dot);

        });

        /\* Muestra la 1era imágen al iniciar pantalla \*/

        mostrarImagen();

    </script>

</body>

</html>

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**