

Fichero: ejercicio1.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio 1</title>

  <style>
    #img_container {
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      height: 500px;
    }

    img {
      max-width: 80%;
      max-height: 80%;
    }
  </style>
  <script src="ejercicio1.js"></script> <!-- Vinculacion con el fichero .js -->
</head>
<body>
  <!-- Todo es creado mediante manipulación del DOM desde JavaScript -->
</body>
</html>
```

Fichero: ejercicio1.js

```
document.addEventListener("DOMContentLoaded", () => {
  /*-- Variables --*/
  const boton = document.createElement("button");
  const img_container = document.createElement("figure");
  const img = document.createElement("img");
  const leyenda = document.createElement("legend");
  const imagenes = [
    "./assets/ejercicio1/img1.jpg",
    "./assets/ejercicio1/img2.jpg",
    "./assets/ejercicio1/img3.jpg",
    "./assets/ejercicio1/img4.jpg"
  ];
  let imgActualIndex = 0;

  /*-- Establece propiedades y añade los elementos al documento --*/
  boton.innerText = "Mostrar imagen";
  img_container.id = "img_container"; // Esto es para que se aplique el estilo CSS
  document.body.append(boton);

  /*-- Controla el evento clic del botón --*/
  boton.addEventListener("click", () => {
    img.src = imagenes[imgActualIndex];
    leyenda.innerText = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/") + 1);

    document.body.append(img_container);
    img_container.append(img);
    img_container.append(leyenda);
    boton.remove();

    document.addEventListener("keydown", handleKeyDown);

    /*-- (EXTRA) Funciones táctiles: Lo he intentado pero no he logrado hacerlo funcionar --*/
    // document.body.addEventListener("touchstart", handleTouchStart);
    // document.body.addEventListener("touchmove", handleTouchMove);
    // document.body.addEventListener("touchend", handleTouchEnd);
  });

  /*-- Controla el evento keyDown del teclado desde document --*/
  function handleKeyDown(event) {
    if (event.code === "ArrowRight") {
```

```

        imgActualIndex = (imgActualIndex + 1) % imagenes.length;
    } else if (event.code === "ArrowLeft") {
        imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;
    }
    /*-- Asigna la imagen nueva --*/
    img.src = imagenes[imgActualIndex];
    leyenda.innerHTML = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/"));
}

/*-- (EXTRA) Controla eventos de las pantallas táctiles (lo he intentado pero no funciona) --*/
let toques;

function handleTouchStart() {
    toques = new Array();
}

function handleTouchMove(event) {
    toques[toques.length] = {x:event.changedTouches[0].clientX, y:event.changedTouches[0].clientY};
}

function handleTouchEnd(event) {
    if (toques.length < 100) { // Si la cantidad de toques es menor que 100
        return;
    }

    let vector = {
        x: toques[toques.length - 1].x - toques[0].x,
        y: toques[toques.length - 1].y - toques[0].y,
    };

    let angulo = Math.atan(vector.y / vector.x);
    // if (vector.y < 0 || vector.x < 0) {

    // }

    // if (vector.y < 0 ? !(vector.x < 0) : vector.x < 0) { //OR exclusivo: o y es negativo, o x es negativo -> 2º y 3er cuadrante
    //     angulo += Math.PI;
    // } else if (vector.y < 0 && vector.x < 0) { // Cuarto cuadrante
    //     angulo += Math.PI * 2;
    // }

```

```
/*-- Determina si ha deslizado a la izquierda o a la derecha --*/
if (angulo > 0) { // Ha deslizado hacia la derecha
    imgActualIndex = (imgActualIndex + 1) % imagenes.length;
} else { // Ha deslizado hacia la izquierda
    imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;
}
// if (angulo > Math.PI / 2) { // Izquierda
//     imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;

/*-- Asigna la imagen nueva --*/
img.src = imagenes[imgActualIndex];
leyenda.innerText = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/"));
}
});
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio 2</title>

  <style>
    * {
      box-sizing: border-box;
    }

    body {
      margin: 0;
    }

    .container {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
    }

    .numpad {
      width: 150px;
      margin: 1em;
      border: 0.5em solid black;
      background-color: #dfdefd;
      border-radius: 6px;
      padding: 0.5em;

      display: grid;
      grid-template-columns: repeat(3, 1fr);
    }

    .numpad input[type=password] {
      width: 80px;
      height: 30px;
      border-radius: 5px;
      grid-column: span 3;
    }

    .numpad button {
      font-size: large;
      margin: 0.2em;
      border-radius: 5px;
      border-width: 2px;
      border-style: outset; /*En firefox si se aprecia, en navegadores chromium*/
    }

    .numpad .numericButton {
      background-color: #78a8f3;
    }

    #validateButton {
      margin-left: 0;
      margin-right: 0;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="numpad">
      <input type="password"/>
      <div class="numericButtons">
        <button class="numericButton">1</button>
        <button class="numericButton">2</button>
        <button class="numericButton">3</button>
        <button class="numericButton">4</button>
        <button class="numericButton">5</button>
        <button class="numericButton">6</button>
        <button class="numericButton">7</button>
        <button class="numericButton">8</button>
        <button class="numericButton">9</button>
        <button class="numericButton">0</button>
      </div>
      <button class="validateButton">Validar</button>
    </div>
  </div>
</body>
</html>
```

```
}

.col-2 {
  grid-column: span 2;
}

.pad-0 {
  padding-left: 0;
  padding-right: 0;
}

.verde {
  color: green;
}

.rojo {
  color: red;
}
</style>
<script src="ejercicio2.js"></script> <!-- Vinculacion con el fichero .js -->
</head>
<body>
  <!-- Todo es creado mediante manipulación del DOM desde JavaScript -->
</body>
</html>
```

Fichero: ejercicio2.js

```
document.addEventListener("DOMContentLoaded", () => {
  /*-- Crea un contenedor --*/
  const container = document.createElement("div");
  container.className = "container";
  document.body.append(container);

  /*-- Crea el div que contendrá el numpad --*/
  const numpad = document.createElement("div");
  numpad.className = "numpad";
  container.append(numpad);

  /*-- Crea un div para mostrar mensajes --*/
  const messageDiv = document.createElement("div");
  container.append(messageDiv);

  // Crea el campo para visualizar los caracteres ocultos
  const passwordField = document.createElement("input");
  passwordField.type = "password";
  passwordField.value = "";
  passwordField.disabled = true;
  // passwordField.style.width = "50px";
  numpad.append(passwordField);

  // Crea un array con los números del 0 al 9
  const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9];

  // Baraja el array de números de forma aleatoria
  for (let i = numbers.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [numbers[i], numbers[j]] = [numbers[j], numbers[i]];
  }

  // Crea los botones de números
  for (let i = 0; i < numbers.length; i++) {
    const button = document.createElement("button");
    button.innerText = numbers[i];
    button.className = "numericButton";
    button.addEventListener("click", () => {
      passwordField.value += button.innerText;
    });
    numpad.append(button);
  }

  // Crea el botón 'C'
  const clearButton = document.createElement("button");
  clearButton.id = "clearButton";
  clearButton.innerText = "C";
  clearButton.addEventListener("click", () => {
    passwordField.value = passwordField.value.slice(0, -1);
  });
  numpad.append(clearButton);

  // Crea el botón 'Validar'
  const validateButton = document.createElement("button");
  validateButton.id = "validateButton";
  validateButton.className = "col-2 pad-0";
  validateButton.innerText = "VALIDAR";
  validateButton.addEventListener("click", () => {
    // Verifica si la contraseña es correcta
  });
});
```

```
const passwordPattern = /^d{4}$/;
if (!passwordPattern.test(passwordField.value)) {
  messageDiv.className = "rojo";
  messageDiv.innerText = "La clave debe ser un número de 4 dígitos.";
  return;
}
if (passwordField.value === "9999") {
  messageDiv.className = "verde";
  messageDiv.innerText = "Clave correcta.";
} else {
  messageDiv.className = "rojo";
  messageDiv.innerText = "Clave incorrecta.";
}
});
numpad.append(validateButton);
});
```


Fichero: ejercicio3.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio 3</title>

  <style>
    .d-inline-block {
      display: inline-block;
    }

    .f-column {
      display: flex;
      flex-direction: column;
    }

    .f-row {
      display: flex;
      flex-direction: row;
    }

    .f-align-items-center {
      align-items: center;
    }

    .f-justify-content-center {
      justify-content: center;
    }

    .t-align-center {
      text-align: center;
    }

    .verde {
      color: green;
    }

    .rojo {
      color: red;
    }

    div {
      padding-bottom: 0.5em;
    }

    button {
      border-style: outset;
      cursor: pointer;
    }

    input[type="submit"] {
      border-style: outset;
      cursor: pointer;
    }

    table {
      border-style: solid;
```

```

        border-width: 1px;
        border-color: #ffc4c4;
    }

    table td {
        border-style: solid;
        border-width: 1px;
        border-color: #ffc4c4;
        padding: 0.3em;
        cursor: pointer;
    }

    @media screen and (max-width: 992px) { /* Esto es para que sea más fácil de pulsar los
    botones en pantallas táctiles de móviles y tablets */
        table td {
            padding: 1em;
        }

        button {
            padding: 0.5em;
        }

        input {
            padding: 0.5em;
        }
    }

    table td:hover {
        background-color: #ccc;
    }

    table td:active {
        background-color: #a9a9a9;
    }

    .seleccionada {
        background-color: #a9a9a9;
    }
</style>
<script src="ejercicio3.js"></script> <!-- Vinculacion con el fichero .js -->
</head>
<body>
    <!-- Todo es creado mediante manipulación del DOM desde JavaScript -->
</body>
</html>

```

Fichero: ejercicio3.js

```
const nPartidos = 14;
document.addEventListener("DOMContentLoaded", () => {
  /*-- Crea un contenedor flex --*/
  const container = document.createElement("div");
  container.className = "f-column f-align-items-center f-justify-content-center";

  /*-- Crea la barra de arriba --*/
  const row1 = document.createElement("form"); // Barra de comprobacion
  row1.id = "barraComprobacion";
  // row1.className = "f-row f-justify-content-center";
  row1.className = "t-align-center";

  const btnComprobar = document.createElement("input");
  btnComprobar.type = "submit";
  btnComprobar.textContent = "COMPROBAR";

  const txtCombinacionGanadora = document.createElement("input");
  txtCombinacionGanadora.type = "text";
  txtCombinacionGanadora.maxLength = nPartidos;
  txtCombinacionGanadora.placeholder = "Combinación ganadora";
  // txtCombinacionGanadora.pattern = /[12Xx]{14}/;
  txtCombinacionGanadora.setAttribute("pattern", "[12Xx]+")

  row1.append(btnComprobar, " ", txtCombinacionGanadora);

  /*-- Crea la quiniela --*/
  const row2 = document.createElement("div"); // Div que contendrá la quiniela en sí
  row2.className = "f-column f-align-items-center";

  // let btnParagraph = document.createElement("p");
  const btnLimpiar = document.createElement("button"); // Botón de limpiar
  btnLimpiar.className = "d-inline-block";
  btnLimpiar.textContent = "LIMPIAR";

  const quiniela = document.createElement("table"); // Quiniela

  row2.append(btnLimpiar, document.createElement("br"), quiniela);

  let celdasSeleccionadas = new Array(); // Array de 14 posiciones que almacenará la celda seleccionada para cada fila
  for (let i = 0; i < nPartidos; i++) {
```

```

let fila = document.createElement("tr");
let uno = document.createElement("td");
let equis = document.createElement("td");
let dos = document.createElement("td");

// celdasSeleccionadas[i] = [uno, equis, dos];

uno.textContent = "1";
equis.textContent = "X";
dos.textContent = "2";

fila.append(uno, equis, dos);

quiniela.append(fila);
}

/*-- Crea un div donde diremos si la combinación es ganadora o no --*/
const row3 = document.createElement("div");
const mensaje = document.createElement("p");
row3.append(mensaje);

/*-- Añade controlador de evento para el clic dentro de la quiniela --*/
quiniela.addEventListener("click", (evento) => {
  /*-- Declaración de variables --*/
  let celda = evento.target.closest("td");
  let fila = evento.target.closest("tr");

  /*-- Comprueba si se ha hecho clic en un td --*/
  if (!celda) return;

  /*-- Elimina la clase CSS "seleccionada" del td que hay actualmente seleccionado --*/
  if (celdasSeleccionadas[fila.rowIndex]) {
    celdasSeleccionadas[fila.rowIndex].className = "";
  }

  /*-- Aplica la clase a la celda seleccionada esta vez --*/
  celda.className = "seleccionada";

  /*-- Guarda la celda seleccionada en el array de celdas seleccionadas --*/
  celdasSeleccionadas[fila.rowIndex] = celda;
});

```

```

/*-- Añade el controlador para el botón de "Comprobar" --*/
btnComprobar.addEventListener("click", (evento) => {
    /*-- Verifica que la combinación ganadora introducida cumple el formato de la expresión regular que le hemos asignado --*/
    if (!txtCombinacionGanadora.checkValidity()) {
        mensaje.textContent = "Por favor, introduce una combinación ganadora en el formato adecuado (1-X-2).";
        mensaje.className = "rojo";
        return;
    }

    /*-- Verifica que el cuadro de texto de la combinación ganadora está completo --*/
    if (txtCombinacionGanadora.value.length != nPartidos) {
        mensaje.textContent = "Por favor, introduce una combinación ganadora de 14 caracteres de longitud en el cuadro de texto.";
        mensaje.className = "rojo";
        return;
    }

    /*-- Descarta que no se haya seleccionado una combinación completa --*/
    if (celdasSeleccionadas.length != nPartidos) {
        mensaje.textContent = "Por favor, rellena los resultados de todos los partidos de la quiniela.";
        mensaje.className = "rojo";
        return;
    }

    /*-- Compara caracter a caracter la combinación con la cadena introducida en el input de la combinación ganadora --*/
    let aciertos = 0;
    let combinacionGanadora = txtCombinacionGanadora.value.toUpperCase();
    for (let i = 0; i < nPartidos; i++) {
        if (celdasSeleccionadas[i].textContent === combinacionGanadora[i]) {
            aciertos++;
        }
    }

    if (aciertos === nPartidos) {
        mensaje.textContent = "¡Enhorabuena! Has acertado los " + aciertos + " partidos.";
        mensaje.className = "verde";
    } else {
        mensaje.textContent = "¡Lo sentimos! Has acertado " + aciertos + " partidos.";
        mensaje.className = "rojo";
    }
});

```

```
/*-- Añade el controlador para el evento submit del botón "Comprobar" --*/
row1.addEventListener("submit", (evento) => {
    evento.preventDefault();
});

/*-- Añade el controlador para el botón de "Limpiar" --*/
btnLimpiar.addEventListener("click", (evento) => {
    celdasSeleccionadas.forEach((celda) => {
        celda.className = "";
    });
    celdasSeleccionadas = new Array(); // Vacía el array por completo
    mensaje.textContent = "";
});

/*-- Añade los elementos al body --*/
container.append(row1, row2, row3);
document.body.append(container);
});
```