

Ejercicios básicos JavaScript 3

- 1) Escribir por consola los números de 1 al 10 usando un for:

Sintaxis:

```
for(inicialización; condición; incremento / decremento) {  
    // instrucciones a repetir  
}
```

- En inicialización normalmente crearemos una variable con let y le daremos el valor de inicio
- En condición se pone una condición que ejecutará el bucle mientras se cumpla esta; normalmente incluye la variable de la inicialización
- Incremento / decremento: normalmente incrementaremos la variable de la inicialización en 1, 2, ...
-

- 2) Escribir por consola los números del 10 al 1 usando un for
- 3) Escribir por consola 0, 5, 10, ... hasta 20 usando un for
- 4) Crear una variable llamada final con el valor 15 y mostrar todos los números entre el 1 y ese valor
- 5) Crear una variable llamada inicio antes del for con el valor 5 y hacer que el for muestre los valores entre el valor de esa variable hasta 10. Para ello podemos dejar vacía la inicialización en la instrucción for:

```
for(;condición; incremento / decremento) {  
    // instrucciones a repetir  
}
```

- 6) Generar 10 números aleatoriamente entre 1 y 50 y escribir por consola todos los números mayores de 25:

```
Math.floor(Math.random() * 50 + 1)
```

- 7) Como el anterior, pero indicar después del for cuantos números eran mayores de 25. Para ello tendremos que crear un contador antes del for e ir incrementándolo cuando se cumpla la condición; al final del for mostraremos su valor
- 8) Generar 15 notas aleatoriamente entre 0 y 10 y decir la nota que se ha sacado, siendo de 0 a 5: suspenso, de 5 a 7: aprobado, de 7 a 9: notable, de 9 a 10: sobresaliente
- 9) Modificar en anterior para que muestre al final el número de notas de cada tipo, para lo que necesitaremos 4 contadores.
- 10) Al pulsar en el botón se debe añadir el nombre escrito en la lista agenda

Ejercicios básicos JavaScript 3

Diagrama de la interfaz de usuario 'Agenda' para el ejercicio 10. Muestra un título 'Agenda', un campo de texto 'Nombre', un botón 'insertar' y un campo de lista. Las etiquetas 'nombre', 'insertar' y 'agenda' están en rojo y tienen flechas que apuntan a sus respectivos elementos en la interfaz.

Para ello haremos lo siguiente:

- Creamos un li:
`const li = document.createElement("li");`
- Añadimos el nombre al li:
`li.textContent = el nombre que hay en el input;`
- Añadimos el li al ul agenda:
`elementoUl.appendChild(li);`

Estaría bien comprobar que no se pongan nombres vacíos, que se vuelva a poner el foco en el cuadro de texto del nombre después de añadirlo y también que la tecla Enter añadiera el nombre:

- Suscribirse al evento keypress en el input nombre
- Comprobar si evt.key es igual a "Enter"

11) Como el anterior pero ahora añadiendo antes del nombre H si es hombre y M si es mujer:

Diagrama de la interfaz de usuario 'Agenda' para el ejercicio 11. Muestra un título 'Agenda', un campo de texto 'Nombre', dos botones de radio 'Hombre' y 'Mujer', un botón 'insertar' y un campo de lista. Las etiquetas 'nombre', 'insertar' y 'agenda' están en rojo y tienen flechas que apuntan a sus respectivos elementos. Las etiquetas 'hombre' y 'mujer' están en rojo y tienen flechas que apuntan a los botones de radio.

12) Como el anterior pero ahora añadiendo una imagen antes del nombre (hombre.png o mujer.png).

Para ello debemos:

- Crear un elemento img de modo similar a como se creó el li

```
const img = document.createElement("img");
```

- Asignar a su propiedad src la imagen adecuada

```
img.src = imagen correspondiente entre comillas
```

- Añadir la imagen dentro del li con appendChild
- Crear un span para el nombre, introducirle el nombre con textContent y luego añadirlo al li con appendChild. Hacemos esto pues no podemos usar textContent en el li porque se cargaría la imagen

Ejercicios básicos JavaScript 3

13) Como el anterior, pero añadiendo un botón de eliminar después de cada nombre, aunque no funcionará en este ejercicio:

- Creamos el li
- Le añadimos la clase d-flex y my-2 con:

```
li.classList.add("d-flex", "my-2");
```

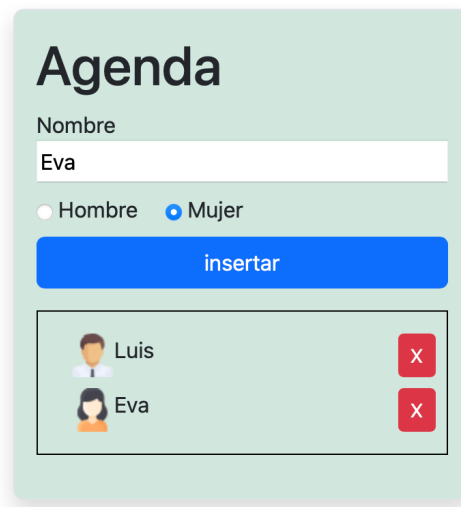
- Después de crear el img, el span y añadirlos al li, creamos un input con document.createElement
- A la propiedad src del input le damos el valor "button"

```
nombreElementoInput.src = "button"
```

- De modo similar le damos el valor "X" a su propiedad src
- Le añadimos todas estas clases:

```
nombreElementoInput.classList.add("btn", "btn-danger", "btn-sm", "ms-4");
```

- Lo añadimos al li con appendChild
- Al span le añadimos la clase flex-grow-1



14) Modificamos el ejercicio anterior para que el botón funcione y elimine la persona donde pulsemos:

- Donde creamos el elemento input le añadimos un listener a la función eliminar

```
nombreElementoInput.addEventListener("click", eliminar);
```

- Creamos la función eliminar con el parámetro evt para saber que persona eliminar

```
// Con removeChild eliminamos el hijo de un elemento (nodo)
// No podemos eliminar currentTarget pues es el botón y queremos eliminar el li al que
// pertenece. Por ello usamos parentNode que devuelve su li padre
function eliminar(evt) {
    document.getElementById("agenda")
        .removeChild(evt.currentTarget.parentNode);
}
```

Ejercicios básicos JavaScript 3

- 15) Añadir un botón Vaciar que borre todos los elementos de la agenda. Para ello basta con poner a cadena vacía "" la propiedad `textContent` del elemento agenda

Diagrama de una interfaz de usuario para una agenda. El formulario se titula "Agenda" y contiene:

- Un campo de texto para el "Nombre".
- Botones de radio para "Hombre" y "Mujer".
- Un botón azul "insertar".
- Un campo de texto para la "agenda".
- Un botón rojo "vaciar".

Las etiquetas "nombre", "insertar", "agenda" y "vaciar" están a la derecha con flechas que apuntan a sus respectivos elementos. Las etiquetas "hombre" y "mujer" están abajo con flechas que apuntan a los botones de radio.

- 16) Antes de vaciar mostraremos un mensaje de confirmación. Para ello podemos usar lo siguiente que devuelve `true` si el usuario pulsó aceptar y `false` si no:

```
const respuesta = window.confirm("Texto a mostrar");
if(respuesta == true) {

}
```