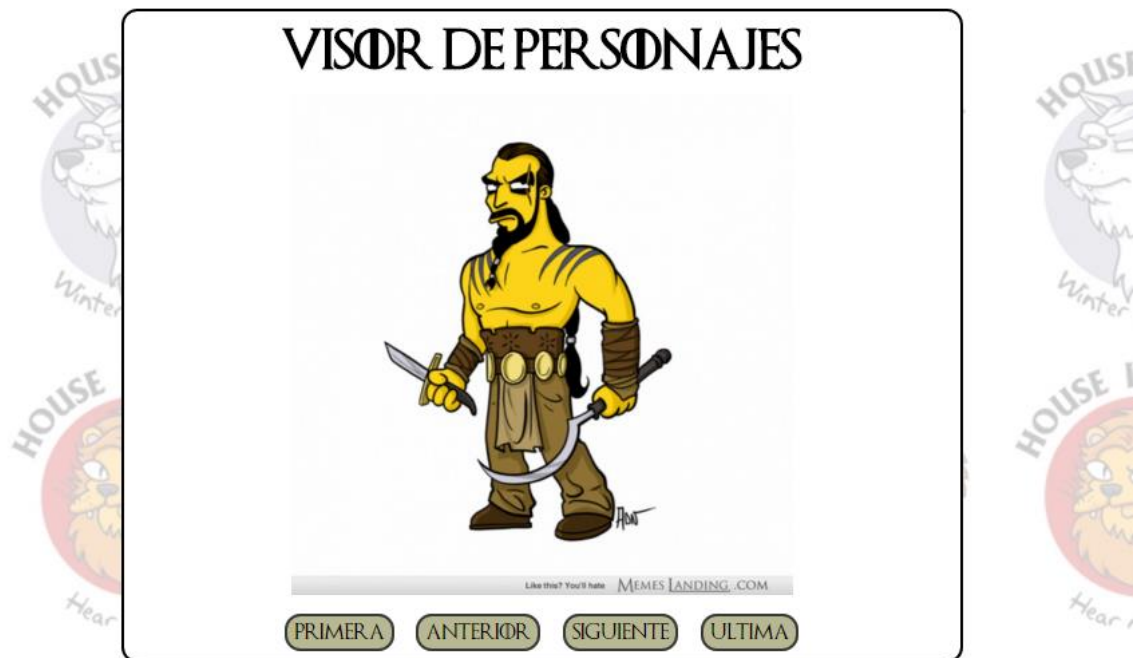


ACTIVIDAD 1

VISOR DE PERSONAJES

Vamos a implementar un visor de imágenes que permite iterar por una serie de viñetas.



Se proporciona la página HTML y la hoja de estilos que hay que usar obligatoriamente. Ninguna puede ser modificada, excepto para incluir el script JavaScript. Las imágenes se cargarán como imagen de fondo de la capa con id **imagen**. Ejemplo de código:

```
document.querySelector('#imagen').style.backgroundImage = 'url(foto1.jpg)';
```

Se definirá en un archivo JavaScript una clase **Galeria** con las siguientes características:

- Propiedades que almacenen la siguiente información:
 - Array que contiene los **nombres** de las imágenes que se van a mostrar.
 - **Índice** de la imagen que se está mostrando actualmente (cursor)
- Métodos:
 - El **constructor** de la clase recibe como único parámetro la cantidad de imágenes que va a manejar la galería. Dado que todas las imágenes tienen el mismo nombre (sólo se diferencian en el número), el constructor generará de manera automática los nombres de todas las imágenes que se van a manejar.
 - Un método que establezca el cursor en una posición **aleatoria** del array, devolviendo el nombre de la imagen que hay en la misma.
 - Un método que establezca el cursor en la **última** posición del array, devolviendo el nombre de la imagen que hay en la misma.
 - Un método que establezca el cursor en la **primera** posición del array, devolviendo el nombre de la imagen que hay en la misma.

- Un método que establezca el cursor en posición **anterior** a la actual, devolviendo el nombre de la imagen que hay en la misma.
- Un método que establezca el cursor en posición **siguiente** a la actual, devolviendo el nombre de la imagen que hay en la misma.

[NOTA1] Al llamar a las imágenes de manera igual, no sería estrictamente necesario el uso del array, pero vamos a mantenerlo, dando a nuestra aplicación la posibilidad de manejar imágenes con distintos nombres.

[NOTA2] Nótese que cuando referenciamos una carpeta desde la hoja de estilos CSS la ruta es relativa a la ubicación de la hoja de estilos, pero cuando accedemos a la propiedad style desde Javascript realmente estamos modificando el estilo in-line (perteneciente al HTML), luego la ruta de acceso será relativa a la ubicación de la página HTML.

Hay que tener en cuenta que:

- Inicialmente se cargará una imagen al azar.
- Si nos encontramos en la primera imagen, los botones *Primera* y *Anterior* se deshabilitarán:



Este comportamiento dejará de producirse cuando cambiamos de imagen.

- Si nos encontramos en la última imagen, los botones *Siguiente* y *Última* se deshabilitarán:



Este comportamiento dejará de producirse cuando cambiamos de imagen.

Para los dos últimos puntos habrá que utilizar los estilos `.habilitado` y `.deshabilitado` que se adjuntan en la hoja de estilos. Ejemplo de código:

```
document.querySelector('#primera').classList.add('deshabilitado');
/* podríamos hacer className="deshabilitado" pero se machacarían otras
posibles clases a las que pueda pertenecer el elemento */
```

[RETO OPCIONAL]

Añade escuchadores de eventos de teclado, de forma que la una pulsación en la flecha derecha (código `ArrowRight`) tenga el mismo efecto que el botón “Siguiente”, y una pulsación en la flecha izquierda (código `ArrowLeft`) tenga el mismo efecto que el botón “Anterior”.