

## Preparación

1. Abrí con el navegador el archivo que veníamos usando, `juego.html`.
2. Abrí el archivo `juego.js` en una pestaña del editor de texto que estés usando. Este está en la carpeta `js` de los recursos descargables.

## Paso 1: Familiarizate con las Variables y Funciones Existentes

### Variable

- El arreglo de instrucciones donde cada elemento será una instrucción del `juego`

```
var instrucciones = []
```

- El arreglo de movimientos donde iremos agregando el último movimiento realizado

```
var movimientos = []
```

- El arreglo de arreglos grilla donde cada número representa a una pieza del rompecabezas

```
var grilla = [  
  [1, 2, 3],  
  [4, 5, 6],  
  [7, 8, 9]  
];
```

- `filaVacía` que guarda la fila de la pieza que se encuentra vacía
- `columnaVacía` que guarda la columna de la pieza que se encuentra vacía

```
var filaVacía = 2;  
var columnaVacía = 2;
```

## Funciones ya implementadas

- `capturarTeclas()` que detecta cuando presionan las teclas para luego poder mover las piezas.
- `mezclarPiezas()` que mezcla las piezas aleatoriamente en el tablero cuando iniciamos el juego.
- `mostrarInstruccionesEnLista(arreglo, idLista)` que muestra los elementos del `arreglo` en el elemento de HTML `idLista`
- `intercambiarPosicionesDOM(pieza1, pieza2)` que intercambia la posición de las piezas en el documento HTML.
- `iniciar()` que comienza el juego, ejecutando las funciones `mezclarPiezas()` y `capturarTeclas()`.

Tené en cuenta que muchas de estas funciones van a depender de otras que tendrás que implementar vos, por lo que no todas funcionarán correctamente desde el principio.

Además, se incluyen funciones incompletas, con comentarios descriptivos de lo que cada función debería hacer. En los distintos pasos vamos a ir viendo qué funciones necesitamos completar para avanzar en el proyecto.

***Nota:** Asegurate de trabajar sobre el archivo `juego.js` y de ir completando las funciones correspondientes en cada etapa.*

## Paso 1: Escribir instrucciones

El objetivo de este primer paso es agregar los textos correspondientes a las instrucciones del juego. Para ello las tendrás que agregar al arreglo `instrucciones` que se encuentra en el código. Podés agregar todas las instrucciones que quieras, teniendo en cuenta que tengan sentido con el objetivo del juego.

**\*\* Recordá que siempre podés chequear lo que acabas de programar desde la consola. Por ejemplo, para ver si las instrucciones las colocaste correctamente en el arreglo, podés ejecutarlo abriendo la consola y chequeando cuales son sus elementos. \*\***

## Paso 2: Mostrar instrucciones

En este paso terminaremos el agregado dinámico de instrucciones al juego. Para eso debemos completar la función `mostrarInstrucciones(instrucciones)`. Tenemos que ir recorriendo cada uno de los elementos del arreglo pasado por parámetro y aplicándole a cada `instrucción` la función ya implementada `mostrarInstrucciónEnLista(instruccion, idLista)`.

**\*\* Abrí el archivo HTML y fijate cuál es el id de la lista dónde deberás agregar las instrucciones. \*\***

## Paso 3: Mostrar último movimiento

El arreglo de `movimientos` es el encargado de guardar todos los movimientos realizados en el juego desde el comienzo hasta el final. Pero esta funcionalidad no está implementada. Es por eso que tenemos que implementar una función que agregue la última `dirección` al arreglo de `movimientos` y, además, ejecutar a continuación la función ya implementada `actualizarUltimoMovimiento` para que se actualice el HTML.

Pista: recordá qué método de los arreglos se utiliza para agregar un nuevo elemento.

**\*\* Para testear que esta función esté correctamente implementada, abrí la consola y ejecutala pasándole como parámetro un código de dirección que podés encontrar en el código. Por**

ejemplo:`nombreQueLeDisteALaFuncion(codigosDireccion.ABAJO)`. Esto debería hacer que aparezca en el HTML una flecha hacia abajo. \*\*

¿Por qué las códigos de dirección?: Es una forma más sencilla de referirse a las direcciones. Sino, tendríamos que usar números como 37, 38, 39, 40 y el código deja de ser declarativo. Estos números están dados por la numeración que reciben las teclas de nuestros teclado.

Mostrar el último movimiento podría haber sido implementado sólo con una variable, pero queremos recordar todos los movimientos realizados para que te animes a implementar otras funcionalidades por tu cuenta. Por ejemplo: ¿cómo sería mostrar los últimos 5 movimientos? ¿O mostrar al final del juego todos los movimientos realizados?