

Ejercicio de Observación directa (Comprensión de Funciones en Javascript)

Ejercicio 1. – Función anónima de autollamada con parámetros fijos y opcionales. (2,5P)

Crea la función llamada **initGame(colorPista, colorPelota)** de autollamada (que se ejecuta inicialmente), que reciba dos parámetros:

- Parámetro 1 llamado **colorPista** , donde recibe una cadena que representa el valor hexadecimal del color del campo. Se usará este parámetro como el color del campo. Este parámetro será obligatorio.
- Parámetro 2 de tipo opcional, llamado **colorPelota** que representa el color de la pelota. Se usará este parámetro como el color de la pelota si la función **initGame** es llamada con él. Si la función **initGame** no es llamada con él se asignará un valor de dicho parámetro por defecto al valor “#fff” (color blanco para la pelota). Este parámetro será un parámetro opcional.

Todo el código del juego deberá de ir dentro de dicha función **initGame**.

Usar la función **initGame** con color de Pista “green” y color de pelota “white”.

Nota: Crea una nueva instancia de tu juego de ping-pong y usa dicha función creada.

Ejercicio 2 .- Objetos como argumentos. (Paso de parámetros a funciones) (2,5P)

Crea la función **gameSettings** que reciba los parámetros del juego como objetos literales. Recibirá un objeto literal de parámetro como el siguiente:

```
{ canvas: document.getElementById("campo") ,  
  anchoPelota: X,  
  altoPelota: Y,  
  avancePelota: 1,  
}
```

Esta función se encargará de inicializar las variables globales:

- **canvas** del campo recibido
- **ancho_canvas** con el ancho de canvas
- **alto_canvas** con el alto del canvas
- **context** con el **canvas.getContext("2d")**
- **x** que es la x inicial de la pelota
- **y** que es la y inicial de la pelota
- **anchoPelota** que es el ancho de la pelota (se recibe como parámetro con valor X, donde representa al ancho, por defecto recibe el valor 20)
- **altoPelota** que es el alto de la pelota (se recibe como parámetro con un valor Y, donde representa el alto de la pelota, por defecto recibe el valor 20)
- **avancePelota** que es el avance de la pelota en cada iteración (por defecto 1, que se incrementa la x e y de la pelota)

Uso de la llamada a dicha función **gameSettings**: (recibe este objeto literal)

```
gameSettings({ anchoPelota: 20,  
               canvas: document.getElementById("campo") ,  
               altoPaltoPelota: 20,  
               avancePelota: 1  
})
```

Deberá ser llamada como se indica arriba.

Esta función **gameSettings** reemplazará a la función **inicializa_parámetros()**.

Nota: Crea una nueva instancia de tu juego de ping-pong y usa dicha función creada.

Ejercicio 3.- Creación de funciones anonimas y valores de retorno (2,5P)

Crea las siguientes funciones:

- a) función anónima que calcula_coordenadas_pelota. Crea una variable global llamada **funcCoordenadasPelota** (crea dicha función anónima y asignala a dicha variable). Al usar dicha función anónima deberá devolver un array con los valores x e y de la pelota. (Función que devuelve un array con la coordenada x e y de la pelota).
Esta variable **funcCoordenadasPelota** reemplaza a la función **calcula_coordenadas_pelota()**, usarla en su lugar.
 - Ejemplo del array devuelto:
[20, 12] donde 20 representa la x de la pelota y 12 representa la y de la pelota.
- b) **Crea una función anónima** que se encarga de dibujar la pelota. Crea una variable global llamada **funcDibujaPelota** que se encargue de dibujar la pelota. Crea dicha función anónima. Esta función anónima reemplaza la función **dibujar_pelota()**.
Esta función no devuelve nada.
Usar la variable **funcDibujaPelota** en lugar de la función **dibujar_pelota()**.

Nota: Crea una nueva instancia de tu juego de ping-pong y usa dicha función creada.

Ejercicio 4.- Funciones como clases. (2,5P)

Crea la siguiente clase llamada **Game**:

Deberá tener lo siguientes **métodos**:

- **dibuja_campo()**
- **calcula_coordenadas_pelota()**
- **dibujaPelota()**
- **dibuja_jugador1()**
- **dibuja_jugador2()**
- **gameLoop()** - bucle del juego
- **initializeGame()**

y como **propiedades**:

posXPelota: posición X de la pelota

posYPelota: posición Y de la pelota

Crea un instancia de Game llamada **pingPongGame** y llama al método **gameLoop()**.

Usa prototype para los métodos **dibuja_jugados1**, **dibuja_jugador2**, y **gameLoop**

NOTA PARA TODOS LOS EJERCICIOS.

Cada ejercicio se entrega en una carpeta llamada “ejercicioX”, donde X representa el número del ejercicio.

Entregar ejercicios con comentarios, libre de errores.

Formatear el código del ejercicio (aplicar formato a dicho código – IMPRESCINDIBLE)

Para los ejercicios 1 a 3 se pide una copia de tu juego de ping-pong con lo que se pide en dicho ejercicio (crear funciones, etc, donde se deberá reemplazar lo creado por alguna de las funciones del juego original de ping-pong seguido en el manual).

Se entrega un archivo comprimido con

“tuPrimerApellido_tuSegundoApellido_tuNombre”.zip/rar

Representa el nombre del archivo a entregar.

Reemplaza tuPrimerApellido , tuSegundoApellido y tuNombre por tus valores.