

Actividades (Solucionadas)

1. Crea un objeto Punto con dos enteros llamados X e Y como propiedades principales. Sus valores deben ser introducidos en el constructor. Crea también un objeto Cuadrado en el que encontremos dos puntos de referencia (esquina superior izquierda e inferior derecha). Añade a este objeto métodos para calcular su área y su perímetro. Por último, crea un array de cinco cuadrados y calcula cual es el más ancho (y muestra su anchura) y cual es el más alto (y muestra su altura).

Solución en fichero dec-019

2. Crea array de objetos alumno. De cada alumno guardaremos: nombre, apellido y nota final. Una vez tengas creadas la lista, completa las tareas que se listan a continuación. En caso de utilizar *callbacks* usa funciones flecha en lugar de funciones convencionales:
 - a) Crea una tabla de HTML para mostrar todos estos alumnos. Debes permitir que la tabla se reordene siempre pulsando en la cabecera de cada columna. Por ejemplo si pulsamos sobre "Apellido" todos los alumnos se mostrarán reordenados por apellidos.
 - b) Sobre la tabla debes mostrar 3 botones con los textos "Mostrar Todos", "Mostrar suspensos" y "Mostrar aprobados". Cada botón filtrará el array original y mostrará en la tabla solo aquellos que se indican en cada botón

Solución en fichero dec-018

3. Crea un objeto `codCuentaBancaria` que represente los movimientos bancarios de un cliente. El objeto guardará: nombre, apellidos y dni pedidos a través de teclado y un saldo inicial de 1000. Implementa dos métodos llamados `ingresar()` y `descontar()` que reciban cantidades y realicen los movimientos. Implementa también un método llamado `imprimeMovimientos()`. Este método debe imprimir los últimos 10 movimientos de la cuenta indicando el tipo y la cantidad de este. En caso de que la cuenta quede tras algún movimiento en saldo negativo, debe quedar bloqueada impidiéndose cualquier descuento posterior.

Solución en fichero dec-012

4. Dado el siguiente código completa la función `hazAjax` para que funcione correctamente.

```
function hazAjax(url) {

}

window.addEventListener("load", function () {
  let req = hazAjax("servidor.php")
  req.then(function (respuesta) {
    console.log("Respuesta recibida: "+respuesta.responseText)
  })
})
```

Solución en fichero dec-080

5. Partimos de la función `dameId` que ejecutará un callback cuando reciba una identificación válida de un sistema de base de datos. Implementa esta función para que simule una petición que tardará dos segundos en terminar y ejecutar el callback pasado. Usa esta función para que asigne una identificación válida a un objeto. Una vez terminado esto, implementa `dameId_promesa` donde en lugar de enviar el callback por parámetro devolvamos una promesa y se a esta a la que pasemos el callback en el momento de su consumo

Solución en fichero dec-081

6. Utiliza un formulario para pedir al usuario un teléfono. El teléfono debe tener 9 dígitos que deben estar en grupos de 3 separados por uno o varios espacios cada uno de ellos. Opcionalmente el usuario puede introducir un prefijo internacional de dos dígitos que comience, o bien por dos ceros o bien con el signo + y esté separado del resto de grupos por espacios. Ejemplos de teléfonos válidos:

```
644 342 384
0034 324 134 222
+34 232 123 332
```

Solución en fichero dec-014

7. Genera una función que retorne strings con dígitos numéricos aleatorios que sigan el siguiente formato `dddd#dddd#dddddd:dd`. Utiliza tu función para inicializar un array de 20 posiciones con strings aleatorios que sigan el formato anterior. Una vez tengas construido tu array de 20 cadenas con el formato anterior debes recorrerlo realizando en cada una de las cadenas las dos siguientes modificaciones:

- Cambia el segundo grupo de 4 dígitos junto con las almohadillas que lo rodean y escribe en su lugar por `--`
- Modifica también el último grupo de dígitos de la forma `.... dd:dd` a `.... dd=dd=`

Tanto en las primeras 20 cadenas originales como las otras 20 modificadas debes mostrarlas usando dos listas `UL` en tu documento. A continuación se muestran algunas cadenas de ejemplo y su equivalente modificada:

2116#1821#95@1721:95	2116#--95@1721=95=
3911#3849#47@3195:08	3911#--47@3195=08=
09810#1355#76@0476:65	09810#--76@0476=65=

Solución en fichero dec-011

8. Determina una expresión regular para validar los siguientes elementos:

- Un DNI (con o sin letra)
- Un número de cuenta (separado o no, con espacios o guiones)

Solución en fichero dec-027