

NOMBRE ALUMNO/A Iván Hernández Fuentes

Curso 2018/2019	ACTIVIDADES:	CALIFICACIÓN
Fecha:	Estructuras definidas por el usuario Grupo: 2º DEW	

MÓDULO	UNIDAD DE TRABAJO	
DEW	UT4	A9
A TENER EN CUENTA:		
1) Antes de comenzar rellene Nombre y Apellidos .		

La tarea consiste en la implementación de varios scripts en lenguaje Javascript. Es obligatorio el uso de las funciones que se detallan

- Queremos generar un script destinado a generar las quinielas aleatorias de toda la temporada. Suponemos que la quiniela solo la componen equipos de primera.

La tabla se generará dinámicamente a partir de un array **equipos** que contiene los nombres de los 20 equipos, definido de manera estática en el script

equipos	Alavés	At. Madrid	..	Villareal
Índice	0	1		19

La tabla **quinielas** se generará dinámicamente a partir de un array **equipos** que contiene los nombres de los 20 equipos, definido de manera estática en el script. La posición $[i][j]$ de la tabla **quinielas** representará el resultado del equipo i contra el equipo j , siendo i y j los índices de los equipos en el array de equipos.

quinielas	-	1	1	X	...	2	← Resultados del equipo 0 (Alavés) como local
	X	-	2	1	...	1	← Resultados del equipo 1 (At.Madrid) como local
	2		-	2	...	1	
	...						
	1	X	2	1	...	-	← Resultados del equipo 19 (Villareal) como local

Por ejemplo, en la tabla anterior, el partido Alavés-Atlético de Madrid genera un "1" (posición $[0][1]$) mientras que el Atlético de Madrid – Alavés genera una "X" (posición $[1][0]$).

La diagonal no genera resultados porque representaría el resultado de que un equipo jugara contra sí mismo.

Se utilizarán las siguientes estructuras:

Clase	Descripción
SIMBOLO	Objeto literal que contiene las representa las constantes 1,X y 2

Función	Recibe	Devuelve	Descripción
generaSimbolo()	-	Simbolo	Genera de manera aleatoria un símbolo (1, X o 2).
muestraEquipos()	equipos:cadena[]	-	Muestra los equipos en una tabla.
generaTabla	equipos:cadena[]	cadena[][]	Genera automáticamente la tabla de equipos (sin introducir resultados en ella).
generaResultados	quinielas:cadena[][]	-	Rellena la tabla quinielas que se pasa como parámetro con símbolos generados de manera aleatoria.
muestraDatos	equipos:cadena[] quinielas:cadena[][]	-	Muestra los datos correspondientes a las quinielas de una temporada. En lugar de mostrar la tabla completa, se preguntará al usuario por el nombre de un equipo local y el nombre de un equipo visitante (se repetirá la petición de datos hasta que ambos equipos existan) y se mostrará el resultado correspondiente a dicho equipo. Tras consultar el resultado se consultará al usuario si desea realizar una nueva consulta.
quinielas	equipos:cadena[]	-	Permite generar y consultar las quinielas de una temporada.

Resultado

Archivo | file:///C:/Users/ivan/Google%20Drive/UAUW/SE%20UNDO%20DE%20UAUW/Desarrollo%20web%20en%20entorno%20de%20desarrollo... | Aulas Virtuales CIFP | FuerteNoticia - Las p... | Yomvi es Movistar+ | Como controlar 2 m... | Administración de us... | Home | FuerteTienda > Panel | Proceso de resolución... »

Elements | Memory | Console | Sources | Audits | Network | Performance | Application | Security | NetBeans | Adblock Plus

top | Filter | Default levels

(index)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	" "	"Fc bar...	"Espany...	"Sevill...	"Atléti...	"Alaves"	"Real M...	"Levant...	"Getafe"	"vallad...	"Girona"	"Celta"	"Eibar"	"R. So...	"Betis"	"Alave...	"Valen...	"Villa...	"Athle...	"Legan...
1	"Fc bar...	" "	"2"	"x"	"1"	"1"	"1"	"1"	"1"	"2"	"1"	"2"	"1"	"x"	"1"	"2"	"2"	"1"	"2"	"1"
2	"Espany...	"x"	" "	"2"	"1"	"2"	"1"	"2"	"1"	"1"	"x"	"1"	"1"	"2"	"2"	"1"	"x"	"x"	"2"	"1"
3	"Sevill...	"2"	"x"	" "	"1"	"1"	"x"	"1"	"1"	"2"	"2"	"1"	"2"	"1"	"1"	"1"	"1"	"1"	"1"	"x"
4	"Atléti...	"1"	"x"	"x"	" "	"1"	"2"	"x"	"x"	"x"	"1"	"1"	"1"	"2"	"x"	"1"	"x"	"x"	"1"	"2"
5	"Alaves"	"x"	"1"	"x"	"x"	"2"	"x"	"1"	"1"	"1"	"1"	"1"	"2"	"x"	"2"	"x"	"x"	"1"	"x"	"2"
6	"Real M...	"2"	"2"	"x"	"1"	" "	"1"	"1"	"1"	"2"	"x"	"x"	"x"	"1"	"x"	"1"	"1"	"x"	"x"	"1"
7	"Levant...	"x"	"1"	"x"	"2"	"2"	" "	"2"	"x"	"2"	"2"	"2"	"x"	"1"	"2"	"1"	"1"	"x"	"2"	"x"
8	"Getafe"	"2"	"x"	"1"	"x"	"1"	"1"	" "	" "	"1"	"2"	"1"	"2"	"1"	"x"	"2"	"2"	"1"	"x"	"x"
9	"vallad...	"1"	"2"	"x"	"1"	"x"	"x"	"x"	"x"	" "	"x"	"1"	"2"	"2"	"1"	"x"	"x"	"x"	"2"	"1"
10	"Girona"	"1"	"2"	"1"	"2"	"x"	"x"	"2"	"x"	" "	"2"	"1"	"2"	"1"	"2"	"1"	"2"	"1"	"2"	"1"
11	"Celta"	"2"	"x"	"x"	"x"	"x"	"2"	"2"	"x"	"2"	" "	"x"	"1"	"x"	"2"	"2"	"x"	"2"	"2"	"2"
12	"Eibar"	"x"	"1"	"x"	"1"	"1"	"x"	"x"	"x"	"x"	"1"	" "	"1"	"x"	"2"	"x"	"2"	"x"	"2"	"1"
13	"R. Soc...	"1"	"1"	"2"	"x"	"2"	"1"	"2"	"1"	"1"	"x"	"1"	"2"	" "	"x"	"x"	"1"	"x"	"x"	"x"
14	"Betis"	"1"	"1"	"2"	"1"	"2"	"1"	"x"	"x"	"2"	"1"	"2"	"2"	"x"	" "	"1"	"2"	"2"	"2"	"x"
15	"Alaves"	"x"	"1"	"x"	"2"	"1"	"2"	"1"	"2"	"2"	"x"	"2"	"1"	"x"	"x"	" "	"1"	"x"	"x"	"x"
16	"Valen...	"1"	"1"	"2"	"1"	"x"	"x"	"1"	"1"	"1"	"1"	"x"	"1"	"2"	" "	"1"	"x"	"x"	"2"	"x"
17	"Villar...	"x"	"x"	"2"	"2"	"x"	"x"	"2"	"1"	"1"	"2"	"2"	"2"	"1"	"1"	"2"	" "	"x"	"x"	"x"
18	"Athlet...	"2"	"1"	"2"	"2"	"1"	"1"	"1"	"x"	"2"	"1"	"x"	"2"	"1"	"1"	"1"	"1"	" "	" "	"2"
19	"Legan...	"2"	"x"	"1"	"1"	"2"	"1"	"1"	"x"	"1"	"2"	"1"	"1"	"1"	"1"	"x"	"1"	"x"	"1"	" "
20	"Rayo"	"2"	"x"	"1"	"1"	"x"	"2"	"1"	"x"	"2"	"1"	"x"	"1"	"1"	"1"	"2"	"1"	"x"	"1"	"2"
21	"Huesca"	"1"	"2"	"x"	"1"	"2"	"x"	"1"	"2"	"1"	"1"	"2"	"1"	"2"	"1"	"1"	"1"	"x"	"x"	"2"

Array(22)

2. Escribe un script donde definas las siguientes estructuras de datos:

Clase Jugador [Definida mediante función constructora]			
Propiedades			
Nombre	Tipo	Visibilidad	Descripción
nombre	cadena	Privada	-
numeroVidas	entero	Privada	-
tienePistola	booleano	Privada	-
tieneGranada	booleano	Privada	-
numeroBalas	entero	Privada	-
Métodos			
Nombre	Parámetros	Devuelve	Descripción
Jugador	-	-	Constructor sin parámetros. Se inicializan todas las propiedades.
Jugador	nombre:cadena numeroVidas:entero tienePistola:booleano tieneGranada:booleano numeroBalas:entero	-	Constructor con parámetros. Si no tiene pistola, el número de balas debería inicializarse a 0.
setPropiedad	propiedad:tipo	-	Establece el valor de la propiedad. Realiza las validaciones oportunas respecto a valores no vacíos y consistencia de tipos, asignando un valor por defecto en su caso.
getPropiedad	-	tipoPropiedad	Devuelve el valor de la propiedad
toString	-	Cadena	Devuelve la información correspondiente al jugador en forma de cadena. Por ejemplo: <i>"El jugador Pepe tiene 3 vidas, tiene pistola con 2 balas y granada"</i> .

Clase Partida [Definida mediante función constructora]			
Propiedades			
Nombre	Tipo	Visibilidad	Descripción
jugadores	[]	Privada	-
Métodos			
Nombre	Parámetros	Devuelve	Descripción
Partida	-	-	Constructor sin parámetros. Se inicializan los datos correspondientes a 5 jugadores que se insertan en el array.
imprime	-	-	Recorre el array y muestra por consola los datos de los jugadores
ruletaRusa	-	-	Escoge al azar uno de los jugadores y establece su número de vidas a 0.

Instancia un objeto de tipo **Partida** y llama de manera secuencial a los métodos **imprime()**, **ruletaRusa()** e **imprime()** de nuevo, comprobando que funcionan correctamente.

Resultado

Elements	Memory	Console	Sources	Audits	Network	Performance	Application	Security	»	⋮	×
top											
Filter											
Default levels											
El jugador Ivan tiene 3 vidas, Si tiene pistola con 5 balas y No tiene granada ejercicio2.html:101											
El jugador Andres tiene 7 vidas, No tiene pistola con 0 balas y Si tiene granada ejercicio2.html:101											
El jugador Joaquin tiene 2 vidas, Si tiene pistola con 20 balas y Si tiene granada ejercicio2.html:101											
Ivan tiene 3 vidas ejercicio2.html:139											
Andres tiene 7 vidas ejercicio2.html:139											
Joaquin tiene 2 vidas ejercicio2.html:139											
En la ruleta rusa salio Joaquin ejercicio2.html:150											
Ivan tiene 3 vidas ejercicio2.html:139											
Andres tiene 7 vidas ejercicio2.html:139											
Joaquin tiene 0 vidas ejercicio2.html:139											

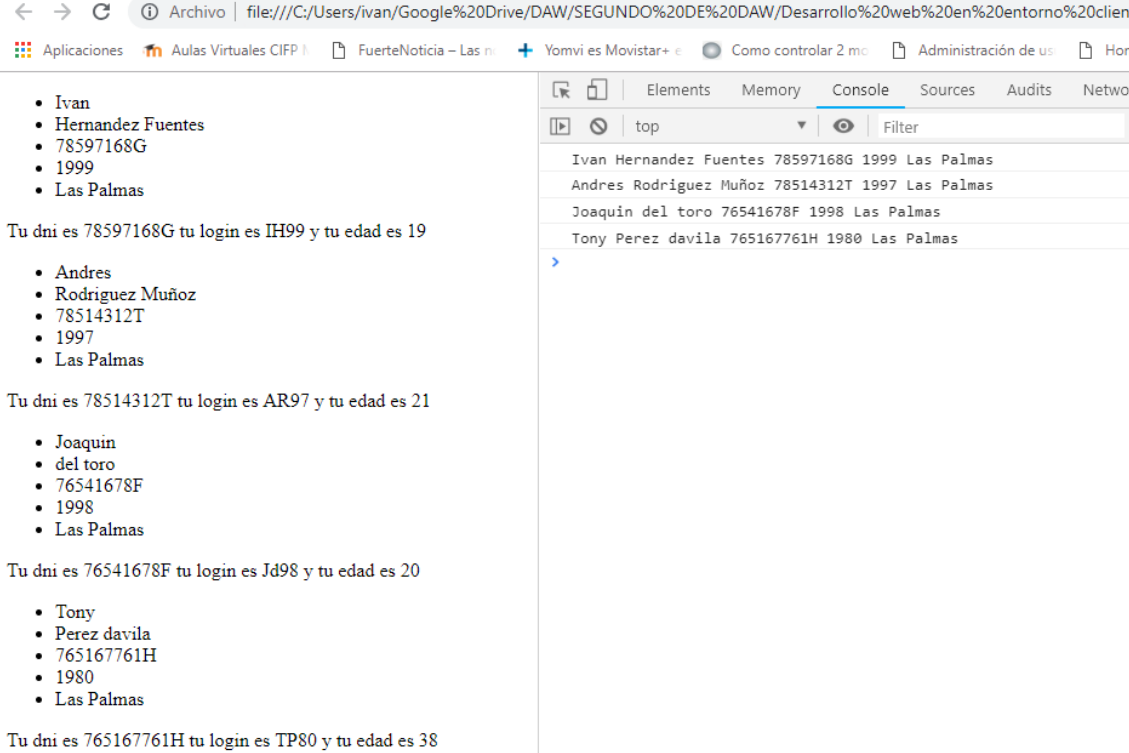
3. Implementa un script que contenga la siguiente estructura de datos:

Clase Usuario [Definida mediante prototipos]			
Propiedades			
Nombre	Tipo	Visibilidad	Descripción
nombre	cadena	Privada	-
apellidos	cadena	Privada	-
dni	cadena	Privada	-
aNacimiento	entero	Privada	-
Provincia	cadena	Privada	-
Métodos			
Nombre	Parámetros	Devuelve	Descripción
Usuario	-	-	Constructor sin parámetros. Se inicializan todas las propiedades.
Usuario	nombre:cadena apellidos:cadena dni:cadena aNacimiento:entero provincia:cadena	-	Constructor con parámetros.
setPropiedad	propiedad:tipo	-	Establece el valor de la propiedad. Realiza las validaciones oportunas respecto a valores no vacíos y consistencia de tipos, asignando un valor por defecto en su caso.
getPropiedad	-	tipoPropiedad	Devuelve el valor de la propiedad
generaLogin	-	cadena	Genera una cadena consistente en la inicial del nombre concatenada con la inicial del primer apellido y el segundo apellido, además de los dos últimos dígitos del año de nacimiento.
getEdad	-	entero	Devuelve la edad del usuario.
toString	-	cadena	Devuelve una cadena que consiste en todas las propiedades concatenadas en una única cadena sin formato.
toHTML	-	cadena	Devuelve una cadena que consiste en todas las propiedades concatenadas en una cadena en formato HTML (puede ser en forma de listado, un párrafo... como se prefiera).

Además de la implementación de la clase, en el script correspondiente:

- Se crearán 4 usuarios de prueba.
- Se mostrarán en la consola de Javascript las propiedades de cada usuario.
- Se mostrarán en la página HTML las propiedades de cada usuario.
- Se mostrará en un listado en la página HTML el DNI de cada usuario seguido de su login y su edad.

Resultado



file:///C:/Users/ivan/Google%20Drive/DAW/SEGUNDO%20DE%20DAW/Desarrollo%20web%20en%20entorno%20clien.

Aplicaciones Aulas Virtuales CIFP FuerteNoticia – Las n... + Yomvi es Movistar+ e Como controlar 2 m... Administración de us... Hor...

- Ivan
- Hernandez Fuentes
- 78597168G
- 1999
- Las Palmas

Tu dni es 78597168G tu login es IH99 y tu edad es 19

- Andres
- Rodriguez Muñoz
- 78514312T
- 1997
- Las Palmas

Tu dni es 78514312T tu login es AR97 y tu edad es 21

- Joaquin
- del toro
- 76541678F
- 1998
- Las Palmas

Tu dni es 76541678F tu login es Jd98 y tu edad es 20

- Tony
- Perez davila
- 765167761H
- 1980
- Las Palmas

Tu dni es 765167761H tu login es TP80 y tu edad es 38

Elements Memory Console Sources Audits Netwo...

top Filter

```

Ivan Hernandez Fuentes 78597168G 1999 Las Palmas
Andres Rodriguez Muñoz 78514312T 1997 Las Palmas
Joaquin del toro 76541678F 1998 Las Palmas
Tony Perez davila 765167761H 1980 Las Palmas
  
```

4. Crea la una pseudoclase llamada **Instituto** que permita crear objetos de este tipo. La clase deberá implementarse mediante prototipos.

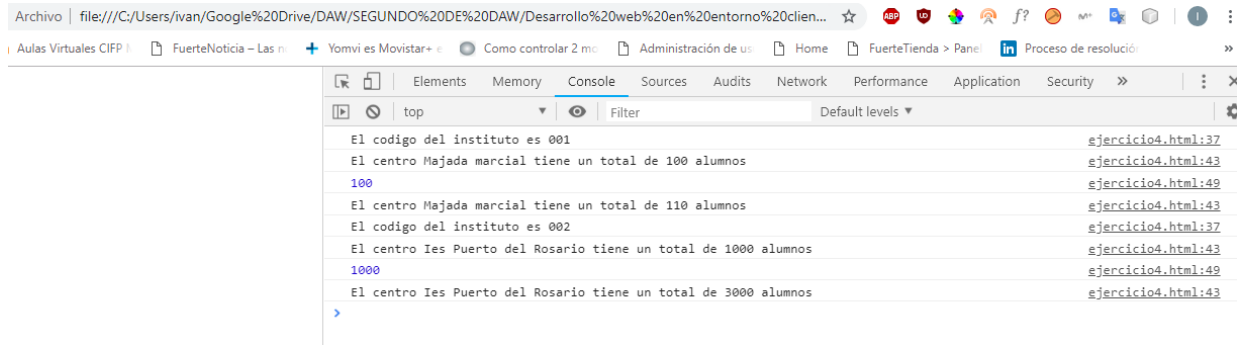
El constructor será de la forma **Instituto(codigoInstituto,nombre,grupos)**, donde grupos es un array simple que contiene las pseudoclases de todos los grupos que tiene el instituto. Para grupo se almacenará el código de grupo y el número de alumnos.

La pseudoclase incluirá los métodos necesarios para:

- Conocer el código del instituto.
- Conocer el número total de alumnos del centro.
- Añadir un nuevo grupo al instituto. Dicho método será de la forma **nuevoGrupo(codigo,nAlumnos)**.

NOTA: El enunciado puede dar lugar a distintas interpretaciones, se da libertad al alumno para que las resuelva como crea más conveniente siempre que se cumplan las especificaciones descritas.

Resultado



- Queremos hacer una aplicación en JavaScript para gestionar edificios con la información de la situación del edificio y de los propietarios de cada piso. Para ello queremos almacenar la siguiente información de cada edificio mediante una clase:

- Calle.
- Número.
- Código postal.
- Plantas del edificio (dentro de cada planta tendremos un número de puertas y para cada puerta almacenaremos el nombre del propietario). Se implementará esta propiedad utilizando un array bidimensional.

Cada vez que instanciamos un edificio le pasaremos la calle, número y código postal como parámetros. Se pide además crear los siguientes métodos:

Nombre	Parámetros	Devuelve	Descripción
agregaPlantasYPuertas	nPlantas:entero nPuertas:entero	-	Recibe el número de plantas que queremos crear en el edificio y el número de puertas por planta. Las plantas se añadirán a las ya creadas en el edificio.
modificaNumero	numero:entero	-	Actualiza el número del edificio
modificaCalle	calle:cadena	-	Modifica el nombre de la calle.
modificaCP	cp:cadena	-	Modifica el CP
getCalle	-	cadena	Devuelve el nombre de la calle
getNumero	-	entero	Devuelve el número del edificio
getCP	-	cadena	Devuelve el CP
agregaPropietario	nombre:cadena planta:cadena puerta:cadena	-	Asigna el propietario nombre al piso identificado por planta y puerta .
imprimePlantas	-	-	Recorre el edificio e imprime por consola los propietarios de cada piso.

Cada vez que se crea un edificio que muestre automáticamente un mensaje del estilo:
Construido nuevo edificio en calle: xxxxxx, nº: xx, CP: xxxxx.

Cada vez que se añade un propietario a un piso de un edificio que muestre un mensaje del estilo:

xxxxxxx es ahora el propietario de la puerta x de la planta x.

Se crearán varias instancias que nos permita comprobar que funciona correctamente, llamando los métodos que hemos definido.

Resultado

