

1. **Crear** un objeto Date, para aplicarle lo siguiente:
 - a. Mostrar la fecha con un alert.
 - b. A partir del texto "July 20 2017", crear un objeto Date y mostrarlo con alert.
 - c. A partir de los datos "2017, 1, 15, 12, 35, 0", generar una fecha y mostrarla con alert.
 - d. Leer los datos del año con getFullYear(), getMonth() y getDate(). Mostrar con alert.
 - e. Incrementa en 6 días la fecha de hoy. Usar setDate() y getDate().
 - f. Calcula la diferencia de días entre hoy y la fecha de tu cumpleaños. Exprésala en segundos y días.
 - g. Compara la fecha de hoy y la de tu cumpleaños, sin horas y comprueba que son diferentes. Muestra el correspondiente mensaje por pantalla.
2. **Hacer** un simple ejercicio html, en el que se llame a 4 funciones desde un botón, para conseguir:
 - a. Abrir una nueva localización
 - b. Actualizar la página
 - c. Abrir una nueva ventana
 - d. Arrancar un temporizador cada 2 segundos que nos muestre la hora que es.
3. **Realizar** un programa html con javascript, en el que se establezca un intervalo de tiempo de 1 segundo para llamar a una función incrementa que sume segundos. Hacer un botón en el que al hacer clic se cancele la ejecución de ese contador y nos muestre por pantalla cuántos segundos han transcurrido

hasta que hemos hecho clic en el botón.

4. **Dado** un array1= [1,2,3] y array2= [4,5,6], realizar las siguientes operaciones con ellos:

- Concatena el array2 y el array1, obteniendo array3. Mostrar el array.
- Elimina el último elemento del array3 y lo devuelves. Mostrar el array.
- Añade el elemento 7 al final del array3. Mostrar el array.
- Elimina el primer elemento del array3 y dí cuál es su longitud. Mostrar el array.
- Añade 0, 1 al principio del array. Muestra longitud y vector.
- Coloca en orden inverso los elementos de un array. Muestra el vector.
- Dados este array: array1=["Hola", "mundo"]. Une los elementos del array con un " " en medio. Además prueba la acción contraria a join, split.

5. Partiendo del siguiente **array** [4,0,3,4,7,3,5,8,1,8,8,0,2,3,1,2,5,7,3,2,5,1], crear un nuevo **array** de elementos del array original sin repetir y ordenado.

6. Crear un objeto, llamado persona:

- Dos propiedades: nombre, edad. Valores: Ana, 45.
- Mostrar un alert con el siguiente mensaje: "Mi nombre es xxx" y "Tengo xxx años".
- Cambiar el nombre a Pepe y añadir propiedad trabajo con valor Informático.
- Añadir un objeto dentro de persona:
 - Address: propiedades: (Inventa los valores)
 - Calle
 - Cp
 - Localidad
- Crear dos métodos: mostrardatos() y cambiarnombre(), que haga exactamente eso y esta última que devuelva el nombre viejo.
- Sustituye persona por this.
- Usa un constructor para crear el objeto persona, invócalo con el valor "Juan", "Pedro" y "María". Llama al método mostrarnombre(), para visualizarlo. Hacer las propiedades privadas.

7. Crear un **prototipo** llamado **coche**(marca, color, annos) con tres parámetros y un objeto del mismo coche1 con el método **new**:

- Tres propiedades: marca, color, años.
- Dos métodos: mostrarmarca_color, cambiar_color
- Crear una variable coche1 con la marca BMW, rojo, 10.
- Añadir un nuevo método al prototipo coche, mostrar_annos

8. Definir la siguiente jerarquía de objetos, haciendo uso de los prototipos de JavaScript:

Objeto Persona con las propiedades nombre, edad y género, y el método obtDetalles(), que

Comentado [UdW1]: Crear una PPT específica de OBJE-TOS y PROTOTIPOS. CAP 8 Y 9

muestra por pantalla las propiedades de la persona.

Objeto Estudiante, que hereda de Persona, e incluye las propiedades curso y grupo y el método registrar().

Objeto Profesor, que hereda de Persona, e incluye las propiedades asignatura y nivel y el método asignar().

Crear los objetos y casos de prueba necesarios para comprobar el correcto funcionamiento de la jerarquía.

9. El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.

El array de letras es:

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H',  
'L', 'C', 'K', 'E'];
```

Por tanto si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A. Con estos datos, elaborar un pequeño script que:

Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado.

En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.

Si el número es válido, se calcula la letra que le corresponde según el método

explicado anteriormente.

Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

10. Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

11. Definir una función que determine si la cadena de texto que se le pasa como parámetro es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha. Ejemplo de palíndromo complejo: "La ruta nos aporotó otro paso natural".

12. Crear las expresiones regulares necesarias para resolver los siguientes puntos:

- Crear una expresión regular valide una fecha en formato "XX/XX/XXXX", donde "X" es un dígito. Probarlo con la expresión: "Nací el 05/04/1982 en Zaragoza."
- Crear una expresión regular que valide una dirección de email. Para simplificar, los valores antes de la @ pueden contener cualquier carácter alfanumérico, y los caracteres . y -, mientras que los valores tras la @

pueden contener caracteres alfanuméricos, y el tipo de dominio puede tener una longitud de 2 o 3 caracteres.

- Dada la siguiente función que de reemplazo de caracteres, reescribirla utilizando expresiones regulares.

```
function escapeHTML(text) {  
  
var replacements = [["@&", "&amp;"], ["'", "&quot;"],  
  
["<", "&lt;"], [ ">", "&gt;"]];  
  
forEach(replacements, function(replace) {  
  
text = text.replace(replace[0], replace[1]);  
  
});  
  
return text;  
  
}
```

- Dados un nombre y un apellido, crear la expresión regular necesaria para mostrarlos en orden inverso y separados por una coma. Por ejemplo, la cadena "John Smith", convertirla en "Smith, John".
- Crear una expresión regular que elimine las etiquetas potencialmente peligrosas (<script>...</script>) y todo su contenido de una cadena HTML.

13. Calcula el número más alto del siguiente *array* identificado por la variable



**Sagrado
Corazón**
Jesuitas • Logroño

DWEC_Tema3 – Write a programII

valores [252,56,33,876,32,985,729,36,184].

14. Crear una expresión regular que permita identificar si un número de teléfono cumple con el formato establecido: 943-852-978, 943 852 978 o 943852978 o bien móvil: 678-789098, 779-567567, 678 678543, 656789789, 777 987987, 754987654.