Ejercicios Unidad 4 – Arrays, funciones y objetos

Realiza cada ejercicio en una carpeta diferente dentro de la carpeta Funciones-Objetos de la unidad 4

1. Ejercicio función flecha

Crea una función flecha que sirva para devolver verdadero o falso si un número es par y falso si no lo es. Demuestra el funcionamiento de dicha función asignándole una variable.

2. Parámetros por defecto

Explica el resultado del siguiente código y el funcionamiento general de los parámetros por defecto.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script>
        function Calcula (numero=1){
            let calculo=numero*numero;
            document.write(`<p>Resultado = ${calculo}`);
        }
    </script>
    <title>Ejercicio de Calculo</title>
</head>
<body>
        <input type="button" onclick="Calcula()" value="Calculo">
        <input type="button" onclick="Calcula(5)" value="Otro Calculo">
 </body>
 </html>
```

3. Función flecha resta

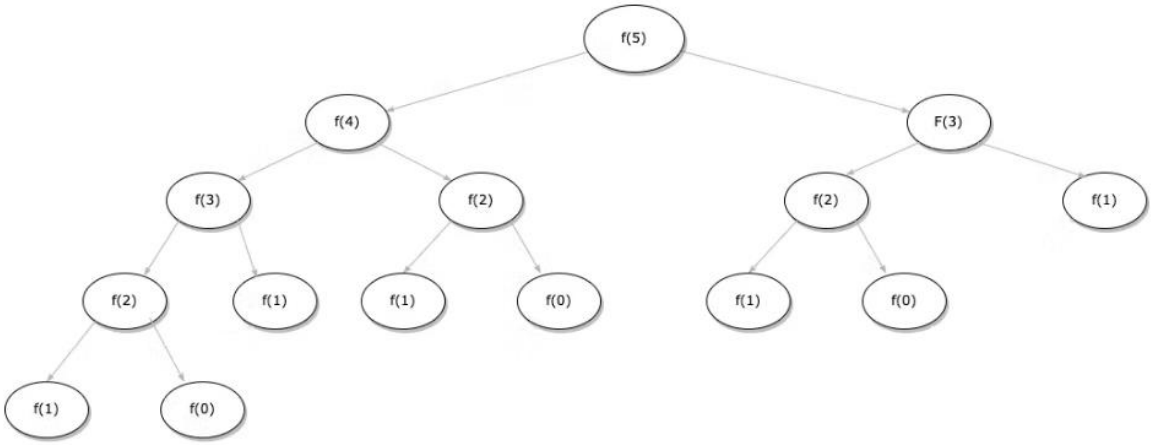
Escribe una función flecha que devuelva la resta de dos números introducidos por el usuario en un único prompt serparados por comas. La función debe recibir dos parámetros. Muestra el resultado con un alert.

4. Función flecha Celsius – Fahrenheit

Escribe dos funciones flecha para pasar un valor dado por el usuario en un input bien de Celsius a Fahrenheit, bien de Fahrenheit a Celsius. La elección se hará mediante un botón.

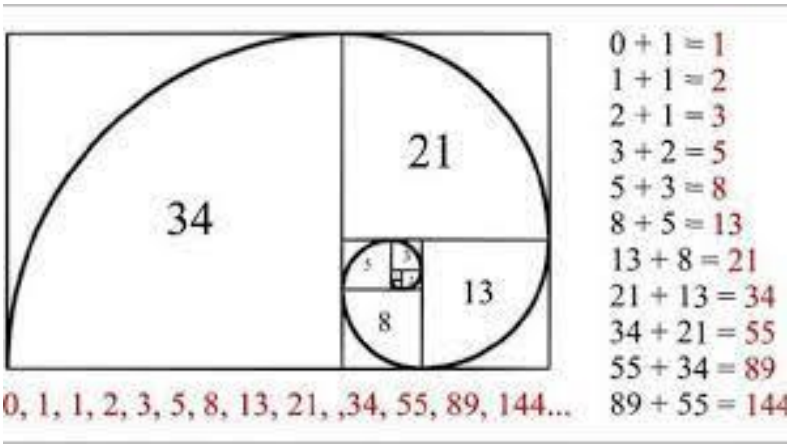
5. Función Recursiva

Un número de Fibonacci, usualmente con notación f(n), es la suma de los dos números Fibonacci que le preceden. Esta sucesión empieza con f(0) = 0, f(1) = 1, f(2) = f(1) + f(0) hasta f(x) = f(x-1) + f(x-2).
Por ejemplo: f(5)



Es decir
F(5)=f(4) + f(3) = f(3) + f(2) + f(2) + f(1) = f(2) + f(1) + f(1) + f(0) + f(1) + f(0) + f(0) + …… = 13

Sucesión de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, ….



f(1)
f(2)
f(3)
f(4)
f(5)
f(6)
f(7)
…

a) Construir una función anónima para obtener el valor fibonacci de un número. Por ejemplo: fibonacci(5) debe devolver 13.
b) Convertir a función flecha
c) Función que construya un array con la sucesión de Fibonacci hasta ese número. (No tiene que ser recursiva)

6. Objeto sencillo

Escribe el código, una línea para cada acción:

1) Crea un objeto user vacío.
2) Agrega la propiedad name con el valor John.
3) Agrega la propiedad surname con el valor Smith.
4) Cambia el valor de name a Pete.
5) Remueve la propiedad name del objeto.

7. Verificar objeto vacío

Escribe la función isEmpty(obj) que devuelva el valor true si el objeto no tiene propiedades, en caso contrario false.

Debería funcionar así:

```
let schedule = {};
alert( isEmpty(schedule) ); // true
schedule["8:30"] = "Hora de levantarse";
alert( isEmpty(schedule) ); // false
```

8. Suma de propiedades del objeto

Crea un objeto ***salarios*** con los salarios de 4 personas.

a) Escribe el código para sumar todos los salarios recorriendo el objeto. Comprueba que funciona
b) Introduce un método llamado ***total*** al objeto que devuelva el total de todos los salarios y comprueba que funciona.

9. Modificación de objeto

Crea un función llamada ***multiplica*** que reciba un objeto (recuerda que se pasan por referencia) y que modifique el objeto multiplicando sus valores numéricos por 2.

Ejemplo de resultado:

```
// Antes de la llamada
let menu = {
  width: 200,
  height: 300,
  title: "Mi menú"
};

multiplica(menu);
```

```
// Después de la llamada
menu = {
  width: 400,
  height: 600,
  title: "Mi menú"
};
```

`multiplica` no necesita devolver nada, solo modificar el objeto.

Usa `typeof` para verificar si la propiedad es numérica.

10. Objeto Calculadora

Crear una **función constructora Calculator** que crea objetos con 3 métodos:

- *read()* pide dos valores usando prompt y los guarda en las propiedades del objeto con los nombres a y b.
- *sum()* devuelve la suma de estas propiedades.
- *mul()* devuelve el producto de la multiplicación de estas propiedades.

11. Uso de clases

Realiza el ejercicio 10 usando la implementación de clases.

12. Herencia con prototipos

Crea el objeto Vehículo con función constructora (marca, modelo, matrícula, incremento, precio)
Las propiedades del objeto serán: marca, modelo, matrícula, color, incremento, precio y velocidad que incialmente siempre valdrá 0
Los métodos serán:

- nombrar: mostrará una cadena “La marca es X y el modelo Y”
- acelerar: que sumará a velocidad *incremento* y devolverá la nueva velocidad;

Crea el objeto Coche que herede de Vehículo. Debe tener dos propiedades nuevas: ruedas con valor y IVA que será definido en la llamada (por ejemplo, valor 21)
También tendrá (además de los métodos mostrar y acelerar) el siguiente método:

- darPrecio: que devolverá el precio más IVA

Mostrar su funcionamiento definiendo instancias de ambos tipos de objeto.

Para terminar muestra una a una todas las propiedades (no las funciones) del objeto Coche (debemos realizar una iteración sobre ellas)

a) solo las propiedades propias del objeto
b) incluye también las propiedades de prototipo

13. Herencia con clases

Realiza el ejercicio 12 con clases

14. Funciones y objetos

Realiza una función flecha que reciba un objeto y añada a dicho objeto una nueva propiedad achura y que a su vez elimine la propiedad altura.
Lista después de la llamada a la función sus propiedades para comprobar el resultado.

15. Funciones y objetos

Realizar una función flecha que devuelva un objeto TODAY con la propiedades day, month y year con la fecha actual.

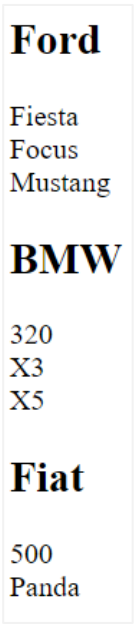
16. Objetos anidados

Mediante literales crea un objeto película para Star Wars con su fecha de rodaje (de tipo fecha, usando el objeto Date) y director. Personajes será otra propiedad de tipo objeto formada por los 3 personajes Han Solo, Leia y Darth Vader (personaje1, personaje2, personaje3).
Usa la función JSON.stringify para mostrar el objeto Película.
Usa la función JSON.stringify para mostrar el objeto Personales.

17. Objetos anidados con arrays

Crea un objeto de modo literal llamado cliente para Juan Cuesta de 30 años. Tendrá las propiedades nombre, apellido, edad y coches.
coches será un array de objetos con las propiedades marca y modelo donde modelo será un array de modelos. Juan Cuesta tendrá los siguientes coches:
Ford modelos Fiesta, Focus, Mustang
BMW modelos 320, X3, X5
Fiat modelo 500, Panda
Una vez creado, mostrar los siguentes elementos:

- Coches de Juan Cuesta (como se muestra en la figura)
- Total de coches de Juan Cuesta
- BMW de Juan Cuesta
- Total de BMW de Juan Cuesta



18. Objetos anidados con arrays

Crea un objeto clase con las propiedades curso, grupo y alumnos donde alumnos será un array de objetos de tipo persona con las propiedades nombre, apellidos y edad.
Crea la clase de 2º DAW con 3 alumnos

19. Array de objetos

Dado el siguiente array de objetos:

```
let cars = [
  {
    "color": "purple",
    "type": "minivan",
    "registration": new Date('2017-01-03'),
    "capacity": 7
  },
  {
    "color": "red",
    "type": "station wagon",
    "registration": new Date('2018-03-03'),
    "capacity": 5
  }
]
```

A. Muestra en HTML (como si fueran los productos del tacón de oro pero sin imagen, puedes usar el color del coche como color de fondo del elemento) cada uno de los coches
B. Introduce usando la función correcta de arrays una furgoneta verde con capacidad 6 al principio del array y muestra de nuevo el HTML
C. Añade una moto roja al final con capacidad 2 y muestra de nuevo el resultado
D. Añade un descapotable azul con capacidad 4 en la segunda posicion y muestra el resultado.
E. Crea una función flecha para mostrar el primer coche rojo que se encuentra
F. Crea una función flecha para mostrar todos los coches rojos que se encuentra
G. Modifica la capacidad de cada coche, para que sea la misma menos un asiento, que se usará para el transportin de los animales.
H. Para cada coche, añade una nueva propiedad size que sea igual a *pequeño* si su capacidad es de menor de 5 y *grande* si es mayor. Usa Array.foreach
I. Ordena los coches por capacidad y muestra el resultado final

20. Función constructora

Repite el ejercicio anterior usando funciones constructoras para los coches.