**Cuestiones Unidad 3: Arrays y objetos definidos por el usuario.**

#### Objetos y Clases

1. Crea un literal de objetos **ordenador**, que almacene el procesador inicializado a i7, la memoria RAM, con valor inicial 12 GB y el disco duro con valor inicial 1TB.

Creamos el literal de objetos

let ordenador ={

Procesado: “i7”,

}

1. Añade al objeto **ordenador** creado en la anterior cuestión, la propiedad SSD con valor inicial 512 GB y elimina la propiedad disco duro. Haz las dos operaciones dos veces: con notación . y con notación [].

Ordenador.SSD=””;

Ordenador[“SSD”]=…;

Delete ordeandor.SSD;

Delete ordenador[“disedio”];

1. Muestra por consola todas las propiedades del objeto **ordenador**. Hazlo de dos formas: utilizando un método de la clase Object y un bucle for..in.

For(prop in ordenador) {

Console.log(prp);

}

1. Indica el resultado de evaluar la siguientes expresiones, después de haber ejecutado las siguientes sentencias:

let pan = {

blanco:6,

deAjo:5,

deSemillas:{

deAvena:3,

deLino:6}

};

delete pan.deAjo;

pan.deQueso = 3;

1. pan["sinSemillas"]["deLino"];

Nos da un error de ejecución no existe.

1. pan.deAjo;

Undefined

1. pan.deSemillas;

deAvena: 3, deLino:6

1. pan.deSemillas["deLino"];

6

1. pan.deQueso;

3

1. Crea la clase **Animal** con la propiedad nombre y el método hablar() que simplemente devolverá el saludo “Rrrrrr”.

Class Animal {

Constructor(nombre) {

this.nombre=nombre;

}

hablar(){

Return “Rrrrr”;

}

}

1. A partir del siguiente código que simula la creación de una clase en JavaScript ES5, genera el código correspondiente para crear la clase según la sintaxis de ES6.

Rectangulo = function(alto, ancho){

this.alto = alto;

this.ancho = ancho;

}

Class Rectangulo {

Constructor(alto, ancho) {

this.alto=alto;

this.ancho=ancho;

}

}

1. Modifica el prototipo de **Rectángulo**, definido en la cuestión anterior, para añadir un método denominado área, que calcule el área del rectángulo.

Rectangulo.prototype.area =function(){

return this.alt \* this.ancho;

}

1. Añade los métodos get y set a la siguiente definición de clase:

class Punto {

constructor(x = 0, y = 0) {

this.\_x = x;

this.\_y = y;

}

}

Get x(){

Return this.\_x;

}

Set x(x){

This.\_x=x;

}

Get y(){

Return this.\_y;

}

Set y(y){

This.\_y=y;

}

1. Crea la clase **Gato**, que herede de la clase Animal, sobrescribiendo el método hablar() para que salude diciendo “Rrrrr” y “miau”.

Class Gato extends Animal {

Costructor(nombre){

Super(nombre);

}

Hablar(){

Return super.hablar() + “miau”;

}

}

1. Crea un método estático para la clase **Rectángulo** que calcule el perímetro. Muestra también cómo harías la llamada al método.

Class Rectangulo {

Constructor(){

}

Static perimetro(){

Return this.

}

}

#### Arrays, operadores rest y spread

1. El siguiente código recorre los elementos de un array. Reescríbelo utilizando el método .forEach() y el bucle for..of.

for (let i = 0; i < lista.length; i++) {

console.log(lista[i]);

}

1. ¿Qué método se utiliza para crear una cadena de texto a partir de los elementos de un array?
   1. join()
   2. concat()
   3. toJoin()
   4. string()
2. ¿Si borramos un elemento de un array con delete (ej. delete array[2];), ¿qué valor tendrá array[2]? ¿Hay otras opciones mejores para eliminar un elemento de un array?
3. Si ejecutamos la sentencia array.splice(2,5,"hola","borrar"), ¿qué ocurrirá?
4. Indica cómo ordenarías el array lista de forma alfabética y de forma numérica, utilizando una función flecha.
5. Partiendo un array de números, utiliza un método que localice la primera posición ocupada por el número 5, sin recorrer todos los elementos. ¿Qué devuelve el método si ese valor no está en ningún elemento del array? Y si lo que queremos es localizar la posición del primer elemento mayor que 5, ¿qué método sería necesario?
6. Partiendo de un array de números, crea un nuevo array a partir de ese, que contenga los valores del array inicial, pero elevados al cuadrado.
7. Partiendo de un array de números, utiliza un método que nos permita obtener la suma de todos sus elementos.
8. Indica el resultado de evaluar la siguientes expresiones, después de haber ejecutado las siguientes sentencias:

let {a, ...b} = {a:1, b:2, c:3, d:4};

let {x, ...t} = {x:0, ...b};

let nombres = Object.keys(b);

1. b.c;
2. nombres[0];
3. b.d;
4. t.c;
5. Indica el resultado de evaluar la siguientes expresiones, después de haber ejecutado las siguientes sentencias:

let pan = {blanco:6, deAjo:5};

let a = pan;

let b = { ...pan };

a.deAjo = 3;

1. pan.deAjo;
2. a === pan;
3. b === a;
4. a.blanco === b.blanco;