Java Script - Objetos

Resumen en base a
https://developer.mozilla.org/es/docs/Web/JavaScript

Objetos

Los objetos en Java Script, pueden verse como estructuras de clave-valor.

Similar a los arrays, se accede a sus datos a través de las propiedades.

Ejemplo:

```
const producto = {
        id: 100,
        descripcion: “Heladera RSD.450”,
        precio: 1500
}
```

Acceso a un objeto. Notación de puntos

Para acceder a un objeto (en base al ejemplo de la diapositiva anterior)

console.log(“id de producto “ + producto.id)

console.log(“descripción del producto ” + producto.descripcion)

console.log(“precio del producto “ + producto.precio)

Acceso a un objeto. Notación corchetes

Se pueden acceder a los datos de los objetos, usando el nombre de la propiedad entre comillas simples o dobles, dentro de un corchete, ejemplo:

console.log(“id de producto “ + producto[“id”])

console.log(“Descripción del producto “ + producto[“descripcion”])

console.log(“Precio del producto “ + producto[“precio”])

Acceso a propiedades con variables

Ejemplo:

```
const persona = {
        nombre : “juan”, edad: 30
}
const nombreVariable = “edad”;
console.log(persona[nombreVariable]);
```

Añadir nuevas propiedades a un objeto de JavaScript

Dado el objeto este objeto:

```
const persona = {
        nombre : “juan”, edad: 30
}
```

Podemos agregar otra propiedad:

```
persona.telefono = “4000-4000”
```

También es posible elimiar una propiedad, ejemplo delete persona.edad

Ejercicio 1

Dada una lista de numero [3,6,-4,8,10,-15,30,0,-12,20]

Calcular, utilizando un objeto para los resultados, cantidad de positivos y negativos, total de positivos y negativos, cantidad y total de números mayores a 8. Mostrar el objeto de los resultados en consola del navegador.

Ejercicio 2

Dada una lista de precios de productos: [500,600,350,1000,200]

Determinar la cantidad y total de precios entre 0 - 500, 501 - 1000 y superiores a 1000.

Utilizar un objeto para mostrar los resultados.

Objetos complejos de JavaScript

Otras de las características de los objetos, es que pueden tener arrays y otros objetos, ejemplo:

```
const persona = {
        nombre: "juan",
        edad: 30,
        telefonos: ["35235","345255","3535235"],
        direccion : {"calle": "San juan", "nro": 1500 }
    }
```

Objetos complejos en JavaScript

También es posible que tengan una función:

```
const persona = {
        nombre: "juan",
        edad: 30,
        telefonos: ["35235","345255","3535235"],
        direccion : {"calle": "San juan", "nro": 1500 },
        cantidadTelefonos: function() {
            return this.telefonos.length;
        }
    }
```

Objetos complejos en JavaScript

A partir de ES6, es posible dejar de usar la palabra reservada function, ejemplo:

```
const persona = {
        nombre: "juan",
        edad: 30,
        telefonos: ["35235","345255","3535235"],
        direccion : {"calle": "San juan", "nro": 1500 },
        cantidadTelefonos() {
            return this.telefonos.length;
        }
    }
```

Ejercicio 1

Dado esta lista de objetos, crear una nueva lista de objetos, pero en vez de tener un array de pagos, debería tener un array con las cuotas en deuda.

[{poliza:123,patente:'AA345FD',cantidad_cuotas:12,pagadas[1,2,3,4]},

{poliza:124,patente:'AA500RR',cantidad_cuotas:6,pagadas[1,2,3,4]},

{poliza:125,patente:'AA200AA',cantidad_cuotas:12,pagadas[1,2,3,4,5,6,7,8,9,10,11,12]},

{poliza:126,patente:'AA300SH',cuotas:8,pagadas[1,2]}]

Mostrar el resultado por consola

Funciones Anónimas y funciones Flecha (arrow function)

Ejemplo de sintáxis:

```
function saludar(alguien) {
    return "Hola" + alguien;
}
console.log(saludar("Carlos"));
const fSaludo = function(alguien) {
    return "Hola desde anonima " + alguien;
}
console.log(fSaludo("Juan"));
const fSaludoArrow = (alguien) => "Hola desde func. arrow " + alguien;
console.log(fSaludoArrow("Pedro"));
```

Palabra reservada class (desde ES6)

La palabra reservada class es solo para sintaxis, comparación con una función:

```
const fSaludo = (alguien) => {this.alguien = alguien;  return this.alguien; }
console.log(fSaludo("Juan"));
class Saludo {
    constructor(alguien){
        this.alguien = alguien;
    }
    fSaludar = () => console.log(this.alguien);
}
const saludo = new Saludo("Pedro");
saludo.fSaludar();
```

Encapsulamiento de las propiedades de una clase. Get and Set

```
class Libro {
    #autor
    constructor(autor) {
        this.#autor = autor;
    }
    get getAutor() { return this.#autor; }
    set setAutor(nombre) { this.#autor = nombre;}
}
const libro = new Libro("Cervantes");
console.log(libro.getAutor);
libro.setAutor = "Borges";
console.log(libro.getAutor);
```

Ejercicio 2

Crear una clase comprobante de factura, con id, fecha, importe, lista de items (cantidad, descripcion, precio unitario) y una función para calcular el total de la factura. Imprimir el resultado en la consola del navegador.