**Curso de JavaScript**

Es case sensitive, Numero no es igual que numero

Es de tipado débil o dinamico

Las variables son del tipo de dato que almacenen, es decir que pueden almacenar distintos tipos de datos.

La sentencias finalizan con ;, es opcional pero se recomienda.

Los bloques finalizan con

Las variables se declarancon let y no con var.

**Declarar variables:**

let numero;

**Inicializar una variable:**

Numero=5;

Let numero=5;

**Declarar Constantes**

Const numero=3.14;

**Tipos de Datos(los principales)**

**Primitivos**

Los que trae el lenguaje

Numeros -> let numero=5; solo existe numero no importa si son decimales

Strings(cadenas) let palabra=’hola’, pueden ser comillas simples o dobles.

Boolean let respuesta=true o false

Undefined indefinido

Null

Symbol

**Orientado a Objetos**

Son estructuras de datos que representan propiedades, valores y acciones que puede realizar el objeto, todos los objetos tienen propiedades o atributos y comportamientos o acciones representados por pares de clave(key): valor(value).

**Ejercicio**

Const computer={

screenSize:17,

model:’MacBook Pro’

}

Const table={

Material:’wood’,

Width:160,

Height:110

}

Const person={

Name:’Juan’,

Age:26,

Sons: [‘Laura’, ‘Diego’]

}

**Imprimir los atributos del objeto**

Console.log(person.name);

Console.log(person.age);

**Tambien se puede acceder a los atributos con un for**

For(const key in person){

Console.log(key);

}

**Para imprimir el valor del array se utiliza of en vez de in, porque**

**Es un array no un objeto.**

For(const son of person.sons){

Console.log(son);

}

**Se puede utilizer Join para separa los datos de un array.**

**Console.log( ${person.sons.join(‘, ‘)});**

**FUNCIONES EN JAVASCRIPT**

Son fragmentos de Código que escribimos para ejecutar una trea y no volver a escribir el mismo código más de una vez.

Nos ayudan a modularizar el código

Las funciones deben realizar una sola tarea.

**Sintaxis de una función**

Function nombreFunción(){

}

Actualmente se declaran así

Const nombreFunción=() ->{

Código a ejecutar

}

**Parametros y Devolver valores**

**Const nombreFuncion=(parámetros)=>{**

**Código a ejecutar**

**}**

**Si solo tenemos una sola instrucción no hace falta utilizar las llaves y el return**

**Const nombreFuncion=(parámetros)=>**

**Parámetros**

**Llamar a una función**

**Saludar();**

For(const key in person){

Console.log(person[key]);

}

**Programación Orientado a Objetos**

Es un paradigma de programación que actualiza la forma de programar.

Const persona{

Nombre=’Juan’,

Apellido=’García’,

Edad=27

}

**Clases – Propiedades**

Son plantillas que se utilizan para crear objetos iguales, cuando se crea un objeto se denomina **instanciar.**

Es necesario crear un constructor y se ejecuta cada vez que creamos un objeto.

**Ejemplo de Clases**

Class Persona{

Constructor(nombre, apellidos, edad){

this.nombre=nombre;

this.apellido=apellido;

this.edad=edad

}

}

This hace referencia al objeto que creamos.

**Métodos**

Los métodos son los que ya existen, ahora las funciones son las que hemos creado. Los objetos pueden tener funciones asociadas a él. En ese caso se denominan Métodos.

Saludar(){

Return ‘Hola, me llamo ${this.nombre} y tengo ${this.edad} años’;

}

**Crear objetos**

Se utiliza la palabra new y el nombre de la clase a utilizar.

Const Juan=new(‘Juan’,’García’,23);

Para acceder a las propiedades se realiza de la siguiente manera:

Juan.nombre

Juan.apellido

Se puede realizar de otra forma utilizando corchetes

Juan[‘nombre’]

Juan[‘saludar’]()

**DOM**

Es toda la estructura html del documento, No es javascript, es una API.

Se utiliza a través de javascipt.

Se utilizan getElementId y otros atributos, para modificar la estructura de un elemento en html.

**Eventos**

Es cualquier cosa que sucede en el documento, los eventos se utilizan para:

El contenido se ha leído

El contenido se ha cargado

El usuario mueve el ratón

El usuario pulsa una tecla

La ventana se ha cerrado y un largo, etc.

**Ejemplo**

Onclick=”saludar()”, no se utiliza

Ahora se realiza de la siguiente manera

(click)=”saludar()”, pero esto solo funciona en angular.

la forma correcta de hacerlo es:

Element.addEventListener(‘event’,callback)