Var es una instrucción nativa del lenguaje, permite declarar variables.

Condicionales y operadores en programación

### Operadores

### Asignación:

var numero1 = 3;

### Incremento y Decremento:

var numero = 5;

++numero;

alert(numero); // numero = 6

var numero = 5;

--numero;

alert(numero); // numero = 4

**Negación**

var visible = true;

alert(!visible); // Muestra "false" y no "true"

var cantidad = 0;

vacio = !cantidad; // vacio = true

cantidad = 2;

vacio = !cantidad; // vacio = false

**And**

&& y su resultado solamente es true si los dos operandos son true

var valor1 = true;

var valor2 = false;

resultado = valor1 && valor2; // resultado = false

valor1 = true;

valor2 = true;

resultado = valor1 && valor2; // resultado = true

**Or**

|| y su resultado es true si alguno de los dos operandos es true

var valor1 = true;

var valor2 = false;

resultado = valor1 || valor2; // resultado = true

valor1 = false;

valor2 = false;

resultado = valor1 || valor2; // resultado = false

**Matemáticos**

var numero1 = 10;

var numero2 = 5;

resultado = numero1 / numero2; // resultado = 2

resultado = 3 + numero1; // resultado = 13

resultado = numero2 – 4; // resultado = 1

resultado = numero1 \* numero 2; // resultado = 50

**Comparacion**

mayor que (>), menor que (<),

**Relacionales**

Los operadores relacionales definidos por JavaScript son idénticos a los que definen las matemáticas: mayor que (>), menor que (<), mayor o igual (>=), menor o igual (<=), igual que (==) y distinto de (!=) comparar negativo.

var numero1 = 3;

var numero2 = 5;

resultado = numero1 > numero2; // resultado = false

resultado = numero1 < numero2; // resultado = true

numero1 = 5;

numero2 = 5;

resultado = numero1 >= numero2; // resultado = true

resultado = numero1 <= numero2; // resultado = true

resultado = numero1 == numero2; // resultado = true

resultado = numero1 != numero2; // resultado = false

**Igualdad**

El operador == se utiliza para **comparar** el valor de dos variables,

var numero1 = 10;

var numero2 = 5;

// El operador "=" asigna valores

var numero1 = 5;

resultado = numero1 = 3; // numero1 = 3 y resultado = 3

**// El operador "==" compara variables**

var numero1 = 5;

resultado = numero1 == 3; // numero1 = 5 y resultado = false

**Comentarios en JS**

//alert: Funcion

// ( ) : Parametros de la función

// "" : Cadenas de texto

**Concatenar:** pegar cadenas

var nombre = "Giovanny";

var apellido = "Calderón";

alert(nombre + " " + apellido)

***Rutas Relativas***

Ejemplo: src="js/piedra.js"

***Truco avanzado: caracteres secretos***

-Salto de línea \n

con , separo los parámetros. peso = prompt("¿Cuál es tu peso en kg?", "100");

***Intrucciones:***

alert muestra avisos

 alert("Ganaste!");

prompt trae cosas del usuario, siempre inserta un dato de tipo texto, así se inserten números.Para solucionar esto se utiliza la Funcion parseInt

**parseInt** convierte cualquier numero que tenga un texto por dentro en un numero total. *parseInt: Pasear como entero*

var peso;

peso = prompt("¿Cuál es tu peso en kg?", "70");

alert( parseInt(peso) + 10);

var peso;

peso = prompt("¿Cuál es tu peso en kg?", "70");

peso = parseInt(peso);

alert( peso + 10);

*El navegador siempre hace primero lo que encuentre entre paréntesis.*

o se puede utilizar **Number /hace lo mismo que parseInt/**

var peso;

peso = prompt("¿Cuál es tu peso en kg?", "70");

peso = Number(peso);

alert( peso + 10);

return

function permite declarar una funcion

**src=””** Fuente del archivo

<script type="text/javascript" src="codigo.js"></script>

pesoEnMarte; **// CamelCase** es una técnica donde se coloca la primera letra en minúscula y cuando se comienza otra palabra la primera letra se coloca en mayúscula.

***strings***

alert("2" + 5 + 8);

258 *// Como el primer dato es un string o texto los demas valores los toma como string*

alert(5 + 8 "2");

132

alert("2" + 5 \* 8);

240 *// porque la multiplicación es mas fuerte que la suma*

**Bloque de instrucciones:**

**if**(opcionUsuario == piedra)  
 {  
        alert("Empate!");  
    }

**Anidar condiciones dentro de condiciones:**

**if**(opcionUsuario == piedra)  
{  
    **if**(opcionMaquina == piedra)  
    {  
        alert("Empate!");  
    }  
    **else** **if**(opcionMaquina == papel)  
    {  
        alert("Perdiste :( ");  
    }  
    **else** **if**(opcionMaquina == tijera)  
    {  
        alert("Ganaste!");  
    }  
}

**Array**

Vectores o Arrays desde Cero, van con corchetes cuadrados [ ]

**var** opciones = ["Piedra", "Papel", "Tijera"];

alert(opciones[1]); //muestra Papel

Donde: “Piedra es igual a 0”

“Papel es igual a 1”

“Tijeras es igual a 2”

**Generar un numero aleatorio:**

//Genera un número aleatorio entre un rango de enteros  
**function** aleatorio(minimo, maximo)  
{  
    **var** numero = Math.floor( Math.random() \* (maximo - minimo + 1) + minimo );  
    **return** numero;  
}

**var** opcionMaquina = aleatorio(0,2);

Math es un objeto

Es el paquete de opciones matemáticas que tiene JS por dentro.

Math.floor

Redondea al mínimo los números decimales

Ejemplo: 2.7 se redondea a 2

Math.ceil \*\*techo\*\*\*

Pasa al siguiente numero los números decimales

Ejemplo: 2.7 se redondea a 3

Math.random

Genera un numero random entre 0 y 1 random= 0 – 1 ejemplo: 0.1, 0.3, 0.9

*Random \* (máximo – mínimo + 1) + min*

Suponiendo que quiero un numero entre 6 y 8

*8 – 6 + 1*

*3*

suponiendo que random genere 0.9

*0.9 \* 3*

*2.7 + 6*

*8.7*

*redondeado hacia abajo da 8*

\*\*\*\*\*-----

Buscar en google Javascript Math MDN(Mozilla developer network)

Tarea: piedra papel tijera lagarto spot

Hacer la interfaz con html en vez de prompt y alert

\*\*\*\*\*-----

**Clase 2**

Para escribir en el HTML se utiliza **document.write**

**var** edad = 28; document.write("Tu edad <br /> es " + edad );

**ln** se utiliza para separa untexto del otro:

document.writeln("escribiendo en HTML");

document.writeln("basico");

escribiendo en HTML basico

Para poder hacer un salto delinea se debería hacer desde el HTML, se pueden agregar etiquetas HTML.

document.writeln("escribiendo en HTML <br/>");

document.writeln("basico");

escribiendo en HTML

basico

var edad = 28;

document.writeln("Tu edad es " + edad);

**MATRICES o VECTORES [ ]**

Son Arrays, colección de información unidas por un mismo concepto.

Un Vector tiene una dimensión

Una Matriz tiene 2 dimensiones.

Menú:

Producto / Ventas / Contacto

0 1 2

*var menu = ["Productos", "Ventas", "Contacto"];*

*document.write(menu[2]);*

En Js para ubicar un punto: [2][0]

Se elije primero X luego Y.

Matriz Multudimensional

var dofa = [["Fuerza", "Oportunidad"], ["Debilidades", "Amenazas"]];

document.write(dofa[0][1]);

Oportunidad

**Funciones - function**

La Funciones solo pueden tener textos o números y deben comenzar con una letra. Se invocan con ()

Declarar una function

**function** explosion()  
{  
    alert("BOOM!!");  
    document.write("<h1>BOOM! Elegiste un area minada :(</h1>");  
}

Invocar una Function

ganaste();

**Objetos del Navegador**

Html tenemos objetos en Document Object Model:

document.write(“Hola mamá!);

write es una función del objeto Document

objeto.funcion o variable(“Parametro”);

Math

Los 3 principales Objetos de Javascript son:

navigator, window, document

**Objetos en JS**

Los Objetos se declara con la primera letra en mayúscula y se declaran con un bloque de código. Terina en **,** y no ;

Al declarar el objeto el bloque termina en **;**

var Pokemon =

{

nombre: "Picachu",

tipo: "Eléctrico"

};

**function** Pokemon(n,v,a)   
{  
    **this**.grito = "Pika!";  
    **this**.nombre = n;  
    **this**.vida = v;  
    **this**.ataque = a;  
    **this**.gritar = **function** ()  
    {  
        alert(**this**.grito);  
    }  
}  
  
**function** inicio()  
{  
    **var** rattata = **new** Pokemon("Rattata", 40, 2);  
    rattata.vida = rattata.vida - 13;  
    nombrePokemon.textContent = rattata.nombre;  
}

//Variables de asignación por valor

//Variables de asignación por referencia

Se pueden crear Objetos dentro de Objetos.

return

this: muchas variables diferentes con diferentes valores pero con la misma estructura

new

**Depuracion de Objetos**

Se puede hacer con *console.log*

*O se escribe directamente el nombre*

***Objeto que contiene un metodo***

*Metodo= función Es una función metida dentro de un objeto*

*Atributo = Variable Cuando la variable es parte de un Objeto se llama Atributo*

\*\*\*\*\*-----

*Tarea*

*Llenar datos de prueba Pokemon*