JScript Teoria Jon Mircha

1. Primitivos: se accede directamente al valor
   1. string
   2. sumber
   3. boolean
   4. null
   5. undefined
   6. NaN
2. Compuestos: se accede a la referencia del valor
   1. object = {}
   2. array = []
   3. function(){}
   4. Class {}
   5. etc.
3. Ordenamiento del código
   1. Importación de Módulos
   2. Declaración de Variables
   3. Declaración de Funciones
   4. Ejecución del Código
4. Palabras reservadas
   1. A: abstract
   2. B: break, boolean, byte
   3. C: case, catch, char, class, const, continue
   4. D: debugger, default, delete, do
   5. E: else, enum, export, extends
   6. F: false, final, float, finally, for, function
   7. G: goto
   8. I: if, implements, import, in,int, instanceof, interface
   9. L: let, long
   10. N: native, new, null
   11. P: package, private, protected, public
   12. R: return
   13. S: short, static, super, switch, synchronized
   14. T: this, throw, throws, try, typeof, transient, true
   15. V: var, volatile, void
   16. W: while, with
   17. Y: yield
5. Notación
   1. Archivo usa **snake\_case**

Mi\_archivo\_javascript.js

* 1. Constantes usa **UPPER\_CASE**

const UNA\_CONSTANTE = “Soy Extremeño”;

const PI = 3.1415926535;

* 1. Clases Usa **UpperCamelCase**

class SerHumano {

constructor (nombre, genero){

this.nombre = nombre

this.genero = genero

}

miNombreEs(){

return “Mi nombre es ${ this.nombre }”;

}

}

* 1. Objetos: Usa **lowerCamelCase**

const unObjecto = {

nombre: “Harnina”,

email = [harnina@gmail.com](mailto:harnina@gmail.com)

}

* 1. Primitivos: Usa **lowerCamelCase**
     1. Let unaCadena = “Hola Mundo”;
     2. const elNumeroNueve = 9;
     3. var unBoolean = true;
  2. Funciones Usa **lowerCamelCase**

function holaMundo(nombre){

Alert(“Hola Mundo” ${nombre});

holamundo(“Harnina”);

}

* 1. Instancias: Usa **lowerCamelCase**
     1. const mesAnnio = new Array()…..

1. das