

Estándar de Codificación - JavaScript.

Propósito	Guiar a los desarrolladores en la escritura de códigos JavaScript.
Estándar de Conteo	<ul style="list-style-type: none"> • Contar cada línea física como un LOC. • No contar líneas vacías y líneas de comentarios. • No contar líneas con una sola llave.
Lenguaje	Define el lenguaje (Idioma) para desarrollar los códigos.
Formato General	Ingles.
Nombrado de Archivos	Define la nomenclatura de nombrado de los archivos físicos donde se desarrollan los códigos.
Formato General – Clase	<ul style="list-style-type: none"> • El nombre debe ser igual al nombre de la clase especificada en el diagrama de clases o encontrada. • Cada clase debe realizarse en un solo archivo de código.
Formato General – Módulo	Nombre del caso de uso (Palabras concatenadas).
Otros	<ul style="list-style-type: none"> • Evitar la mezcla de códigos JavaScript con códigos HTML5, a menos que sea absolutamente necesario. • Cada bloque de código debe de estar escrito en archivos independientes.
Encabezados	Define la estructura de los encabezados que deben de tener los archivos donde se desarrollan los códigos.
Formato General	<pre> /***** Name: Autor name: Modification autor name: Creation date: Modification date: Description: *****/ </pre>
Estructura de Contenidos	Define, de manera resumida, la estructura de los contenidos de archivos de códigos.
Formato General – Clase	<pre> //Declaración de clases //Declaración de atributos/propiedades //Declaración de comportamientos/métodos </pre>
Formato General – Módulo	<pre> //Declaraciones de variables //Declaraciones de funciones </pre>
Instrucciones de Reúso	Define como se prepara el código para reúso. Provee la declaración de formatos, valores, tipos y límites de parámetros. Provee alertas de valores no permitidos, condiciones de sobrecarga u otras condiciones que puedan resultar operaciones con resultados inapropiados.
Formato General	<pre> function getMenuControls(vidUserType) //Purpose: It gets the user menú controls //Limitations: The current user must be a valid user type //Returns a empty or HTML5 code (menu controls) string </pre>
Identificadores	Define de manera descriptiva los nombres de todas las variables, constantes y otros identificadores. Evitar las abreviaturas o variables de una sola letra.

Formato General	<pre> function clspUser(){ //Principal class name function clscUser(){ //Collection class name this.idUser; //Attributes/properties (Variable) class this.getUserName=function(){ //Behavior/Method class this.verifyPassword=function(vpassword){}; //Behavior/Method (Parameter) class function showDataUser(){ //Function var vuserName = "ecabrera"; //Variable var vcivilStatus= new clspCivilStatus();//Variable const cPi=3.14159; //Constant </pre>
Comentarios	Define la forma de documentar códigos para que los lectores puedan comprender las operaciones. Los comentarios deben de explicar tanto el propósito como las conductas de los códigos.
Buenos Comentarios	<pre> this.idIndividual; //Set/get the individual identifier. this.addToDataBase=function() //It adds a new individual to database while (vindex<=vrow.length) //Have all records been read? var vsum = 0; //Set the variable sum to zero </pre>
Malos Comentarios	<pre> this.idIndividual; //Individual identifier. this.addToDataBase=function() //It adds a individual while (vindex<=vrow.length) //Verifies if is end rows var vsum = 0; //Sum is equal to zero </pre>
Espacios en Blanco	Escribir códigos con suficiente espacio de tal forma que no estén amontonados. Separar cada sección de código con al menos un espacio.
Formato General	<pre> function showStatesList() //It shows states data list { //It gets states list in JSON format (string) var vcadenaJSON=xajax.call('showStatesList', {mode:'synchronous'}); //It gets states list in JSON format (object) var vJSON=jQuery.parseJSON(vcadenaJSON); //It sets data to state control \$("#cmbstate").kendoDropDownList({ dataTextField: "text", dataValueField: "value", dataSource: vJSON, index: 0, change: showMunicipalitiesList }); //It sets state control vcmbState=\$("#cmbstate").data("kendoDropDownList"); } function deletePhoneCustomer() //It deletes a phone customer into database { </pre>

	<pre> if (confirm("¿Está seguro que desea eliminar el teléfono?")){ //does the user want to delete the phone? //It deletes the phone customer into database xajax_deletePhoneCustomer(vidCustomer, vidPhoneCustomer); } else{ alert("La eliminación del teléfono ha sido cancelada"); } } </pre>
Sangría	Define la forma de poner sangrías en cada nivel de llaves desde su apertura. La apertura y cierre de llaves deben de estar alineadas unas con otras.
Formato General	<pre> function recordCustomer() //It records/updates a customer into database { if (viBIdentifier==0){ //is the record a new customer? if(validateCustomerFormFields()){ //are customer field data valid? //It records a customer into database xajax_recordCustomer(xajax.getFormValues("vcustomer")); } else{ alert("El registro de los datos del cliente no puede ser realizado, no tiene los datos necesarios, complete los datos"); } } else{ if(validateCustomerFormFields()){ //are customer field data valid? //It updates a customer into database xajax_updateCustomer(xajax.getFormValues("vcustomer"), viBIdentifier); } else{ alert("La modificación de los datos del cliente no puede ser realizado, no tiene los datos necesarios, complete los datos"); } } } </pre>
Escritura	Define la forma de escritura de las palabras que componen las variables y las constantes.
Formato General	<pre> const cPi = 3.14159; var varea= cPi * (vradio * vradio); vtotalArea+=varea; </pre>