Definición de Sintaxis

La sintaxis en informática consiste en las normas a las que debemos atenernos para darle las instrucciones a un programa u ordenador. En otras palabras, sobre cómo debemos organizar las instrucciones para transmitírselas a un ordenador y generar en él una respuesta adecuada.

Los ordenadores desgraciadamente no entienden igual que una persona, y a veces puede ser complicado hacer que haga algo que queremos. Para ejecutar tareas es para lo que tienen un conjunto de instrucciones que deben serle transmitidas de acuerdo a unas reglas determinadas para que las entienda y las ejecute. Desgraciadamente, al igual que existen muchas lenguas en el mundo, existen muchos programas con diferentes instrucciones y con diferentes normas de sintaxis, y aunque el código binario es universal, debido a la imposibilidad para el hombre de trabajar con él (trabajar con cadenas de miles de millones de ceros y unos no es factible), debe usar alguno de estos programas para interactuar con el ordenador, y atenerse entonces a sus reglas y a su sintaxis.

En general la sintaxis cobra especial importancia en los lenguajes de programación (que no son otra cosa que programas usados para crear otros programas más complejos). Con ellos los programadores les pueden dar instrucciones al ordenador básicas para que este realice cosas más complejas, además de permitir traducir esas instrucciones a código binario. Cualquier error en la sintaxis anula la orden y la hace incomprensible para el ordenador, y por tanto impide ejecutarla. Los primeros lenguajes de programación nacieron con las primeras computadoras, y generalmente solo admitían un par de órdenes básicas, pero han ido perfeccionándose con el tiempo. Del antiguo COBOL, nació el BASIC, el PASCAL, el C, el código HTML... cada vez con más capacidades y más sencillos de usar para los programadores.

Lo mismo sucedió con los sistemas operativos y los programas. Con el tiempo y la llegada de los entornos gráficos, los viejos sistemas operativos de consola (donde dabas las órdenes directamente desde el teclado) fueron substituidos por entornos manejados por ratón y/o métodos táctiles. De todos modos la mecánica viene a ser la misma: nosotros le damos una orden al ordenador mediante (por ejemplo) un click sobre una carpeta del ratón en el botón copiar, este lo traduce como 'Copia' la 'Carpeta X' que está en el sitio 'Y' al sistema operativo, y este se lo transmite al ordenador, que lo traduce a código fuente y lo ejecuta. Notad que aunque en este caso nosotros no usemos una sintaxis como tal (solo cliqueamos), el sistema operativo si traduce nuestra acción a una sintaxis adecuada para que el pueda ejecutarla. Si no fuese así, el ordenador no entendería nuestra orden y daría error.

Generalmente la sintaxis en programas suele recurrir a un patrón:

ACCION + OBJETOS IMPLICADOS EN LA ACCION + PARAMETROS

Así, la acción podría ser copiar, formatear, pegar... los objetos serían los archivos o el contenido a copiar, borrar... y los parámetros nos permitirían configurar esa acción, por ejemplo si queremos sobrescribir, borrar definitivamente... etc...

Sin embargo existen miles de programas e incluso de sistemas operativos, y no siempre se sigue esta lógica. Hay incluso programas que intentan comprender las órdenes implícitas en una conversación normal con una persona, pero aún en desarrollo. Incluso existen métodos experimentales que usan órdenes corporales o emitidas mediante ondas cerebrales. Parece que en un futuro nos podremos olvidar definitivamente de la sintaxis, aunque para el ordenador esta siga siendo necesaria.

... Fuente <http://www.mastermagazine.info/termino/6680.php>

La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

* No se tienen en cuenta los espacios en blanco y las nuevas líneas: como sucede con XHTML, el intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo mejor (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)
* Se distinguen las mayúsculas y minúsculas: al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. En cambio, si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona.
* No se define el tipo de las variables: al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
* No es necesario terminar cada sentencia con el carácter de punto y coma (;): en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter ;. Aunque JavaScript no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).
* Se pueden incluir comentarios: los comentarios se utilizan para añadir información en el código fuente del programa. Aunque el contenido de los comentarios no se visualiza por pantalla, si que se envía al navegador del usuario junto con el resto del script, por lo que es necesario extremar las precauciones sobre la información incluida en los comentarios.

JavaScript define dos tipos de comentarios: los de una sola línea y los que ocupan varias líneas.

Ejemplo de comentario de una sola línea:

// a continuación se muestra un mensaje

alert("mensaje de prueba");

Los comentarios de una sola línea se definen añadiendo dos barras oblicuas (//) al principio de la línea.

Ejemplo de comentario de varias líneas:

/\* Los comentarios de varias líneas son muy útiles

cuando se necesita incluir bastante información

en los comentarios \*/

alert("mensaje de prueba");

Los comentarios multilínea se definen encerrando el texto del comentario entre los símbolos /\* y \*/.