

PROGRAMA (*PROGRAM*):

Un conjunto de instrucciones (o sentencias) que describen alguna aplicación o actividad ejecutada en una computadora.

LENGUAJE DE PROGRAMACIÓN (*PROGRAMMING LANGUAGE*):

Notación utilizada por los programadores para escribir programas. Un lenguaje tiene una sintaxis (las palabras y símbolos utilizadas para escribir códigos de programa), una gramática (las reglas que definen una secuencia de palabras y símbolos significativos y correctos) y semántica. JavaScript es un lenguaje de Programación.

ALGORITMO (*ALGORITHM*):

Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan y especifica el orden en que se ejecutan estas acciones. Los algoritmos ayudan al programador a planificar un programa antes de su escritura en un lenguaje de programación.

APLICACIÓN (*APPLICATION*):

Programa autónomo JavaScript tal como cualquier programa escrito utilizando un lenguaje de alto nivel. Las aplicaciones se pueden ejecutar desde cualquier computadora con un intérprete de JavaScript. Se utiliza como sinónimo de programa.

TIPO DE DATOS (*DATA TYPE*):

Un tipo particular de elemento de datos, definido por los valores que puede tomar, el lenguaje de programación utilizado o las operaciones que se pueden realizar en él. En sí, los tipos de datos son elementos que se utilizan para definir variables. JavaScript permite trabajar con los siguientes tipos de datos **string**, **number**, **bigint**, **boolean**, **undefined**, **symbol**, y **null**.

TIPO DE DATO PRIMITIVO (PRIMITIVE DATA TYPE):

Un valor primitivo o tipo de datos primitivo son datos que no son un objeto y no tienen métodos. Los tipos de datos primitivos en JavaScript son **int**, **bigInt**, **float**, **double**, **boolean** y **char**.

BOOLEAN (BOOLEAN, LÓGICO):

Tipos primitivos de datos en JavaScript. El tipo boolean puede tomar sólo dos valores: **true** (*verdadero*) y **false** (*falso*).

CLASE:

Colección encapsulada de datos y operaciones que actúan sobre los datos. El concepto de clase es fundamental en programación orientada a objetos. Una clase consta de métodos y datos. Los métodos de una clase definen el conjunto de operaciones permitidas sobre los datos de una clase (sus atributos). Una clase puede tener muchas instancias de la clase u objetos.

VARIABLE LOCAL:

Una variable utilizada solo dentro de la rutina o función en la que está definida. Cuando la función finaliza y el control vuelve a la parte del programa que la llamó, las variables locales dejan de existir.

VARIABLE GLOBAL (GLOBAL VARIABLE):

Una variable con alcance global, lo que significa que es visible y accesible en todo el programa.

COMENTARIO (COMMENT):

Trozo de texto que tienen como objetivo documentar el programa y mostrar cómo se ha construido. Los comentarios no son sentencias de programación y son ignorados por el intérprete. En JavaScript los comentarios están precedidos por dos barras (//) en una línea o encerrados entre /* y */ en múltiples líneas.

INTERPRETACIÓN:

Acción en la que las instrucciones de un programa no son ejecutadas directamente por la máquina de destino, sino que son leídas y ejecutadas por algún otro programa. JavaScript es un lenguaje interpretado, no compilado.

COMPILACIÓN (*COMPILING*):

Nombre dado al proceso de traducción de un lenguaje de programación de alto nivel a un lenguaje de nivel inferior para crear un programa ejecutable. En otras palabras, la compilación traduce código que es legible para los seres humanos y lo hace ejecutable para los procesadores, en una expresión de código que no es legible para los seres humanos.

EL FLUJO DE CONTROL:

Es el orden en el que se ejecutan o evalúan sentencias, instrucciones o llamadas de funciones individuales de un programa.

CONSTRUCTOR:

Método especial utilizado para inicializar el estado de un nuevo objeto. El constructor permite crear objetos utilizando el operador **new**. El constructor tiene exactamente el mismo nombre que el objeto que lo contiene. Los constructores se pueden sobrecargar con el objetivo de facilitar la construcción de objetos con diferentes tipos de valores iniciales.

CONSTRUCTOR POR DEFECTO (*DEFAULT CONSTRUCTOR*):

Constructor que no tiene parámetros y sirve para inicializar un objeto.

DISEÑO :

Actividad de definir como se debe estructurar e implementar un programa.

FUNCIÓN :

Construcción matemática a la que se pueden aplicar valores y que devuelve un resultado.

HTML (*Hypertext Markup Language*):

Lenguaje de 'script' o de marcas para diseñar páginas Web para creación y compartición de documentos electrónicos integrados preparados para multimedia e Internet.

ETIQUETA (*tag*):

Una instrucción HTML que indica a un navegador Web como visualizar un documento. Las etiquetas se encierran entre corchetes tales como `<html>`, `<i>`, ``, y `</html>`.

IDE:

Software para ayudar a los programadores a escribir código eficientemente.

IMPLEMENTACIÓN (*IMPLEMENTATION*):

La actividad de escribir, compilar, probar y depurar el código de un programa.

INSTANCIA (*INSTANCE*):

Objeto de una clase.

MANEJADOR DE SUCESOS (*EVENT HANDLER*):

Un método en el que el objeto "oyente" se ha diseñado para hacer algún proceso especificado cuando ocurre un suceso determinado.

MARCO DE TRABAJO (*FRAMEWORK*):

Un framework es un conjunto de funcionalidades que dan solución y facilitan la ejecución de una serie de instrucciones.

OBJETO INSTANCIA (*INSTANCE OBJECT*):

Una instancia, en la programación orientada a objetos (OOP), es una realización específica de cualquier objeto. La creación de una instancia realizada se denomina instanciación. Cada vez que se ejecuta un programa, es una instancia de ese programa. En los lenguajes que crean objetos a partir de clases, un objeto es una instanciación de una clase.

OYENTE DE SUCESOS (*EVENT LISTENER*):

El objeto que recibe y maneja el suceso.

PALABRA CLAVE, RESERVADA (*keyword*):

En JavaScript, una palabra clave (o palabra reservada) es una palabra definida como parte del lenguaje de programación. Un nombre de palabra reservada no se puede utilizar para ningún otro propósito.

MÉTODO (*METHOD*):

Una colección de sentencias que se agrupan para ejecutar una operación.

OBJETO (*OBJECT*):

Véase *instancia*. Una instancia de una clase específica. En general, se puede construir cualquier número de objetos a partir de una clase.

OPERADOR (*OPERATOR*):

Operaciones para valores de tipos primitivos de datos. Ejemplos de operadores son +, -, *, / y %.

**PARÁMETRO (*PARAMETER*):**

Los parámetros formales se especifican en una declaración de un método en una llamada a un método.