

alcance

Define la visibilidad y vida de una variable, procedimiento u objeto. Por ejemplo, una variable declarada **Public** está disponible para todos los procedimientos en todos los módulos de todas las aplicaciones, a menos que **Option Private Module** esté en efecto. Cuando **Option Private Module** está en efecto, el módulo es privado y por lo tanto no es visible a los proyectos referidos. Las variables declaradas en procedimientos son visibles solamente dentro del procedimiento y pierden su valor entre llamadas a menos que se declaren **Static**.

aplicación host

Cualquier aplicación que acepta el uso de Visual Basic, Edición para aplicaciones. Por ejemplo, Microsoft Excel, Microsoft Project, etc.

aplicación

Colección de código y elementos visuales que trabajan juntos como un programa único. Los programadores puede desarrollar y ejecutar aplicaciones dentro del entorno de programación, mientras que los usuarios normalmente ejecuta aplicaciones como archivos ejecutables fuera del entorno de programación.

archivo de recursos

Archivo en un proyecto de Visual Basic con una extensión de nombre de archivo .res que puede contener mapas de bits, cadenas de texto usuario otros datos. Almacenando estos datos en un archivo separado, puede cambiar la información sin modificar el código. Sólo se puede asociar un archivo de recursos con un proyecto.

archivo ejecutable

Una aplicación basada en Windows que se puede ejecutar fuera del entorno de programación. Un archivo ejecutable tiene una extensión de nombre de archivo .exe.

argumento

Constante, una variable o una expresión pasadas a un procedimiento.

argumentos con nombre

Argumentos con nombres predefinidos en la biblioteca de objetos. En lugar de dar valores a los argumentos en el orden esperado por la sintaxis, puede utilizar argumentos con nombre para asignar valores en cualquier orden. Por ejemplo, suponga que un método acepta tres argumentos:

HaceAlgo *nombrearg1*, *nombrearg2*, *nombrearg3*

Asignando valores a los argumentos con nombre, puede utilizar la siguiente instrucción:

HaceAlgo *nombrearg3* := 4, *nombrearg2* := 5, nomb

Observe que los argumentos no necesitan estar en el orden normal.

biblioteca de objetos

Un archivo con una extensión de nombre de archivo .OLB que proporciona a los controladores de Automatización OLE (como Visual Basic) información sobre los objetos de Automatización OLE disponibles. Puede utilizar el **Examinador de objetos** para examinar el contenido de una biblioteca de objetos y obtener información sobre los objetos que proporciona.

biblioteca de tipos

Un archivo o un componente incluido en un archivo que contiene descripciones estándar de Automatización OLE de objetos, propiedades y métodos expuestos. Los archivos de biblioteca de objetos (.olb) contienen bibliotecas de tipos.

biblioteca de vínculos dinámicos (DLL)

Una biblioteca de rutinas cargadas y vinculadas en las aplicaciones durante el tiempo de ejecución. Los archivos DLLs se crean con otros lenguajes de programación como C, MASM o FORTRAN.

código de caracteres

Número que representa un carácter en particular de un juego de caracteres, como el juego de caracteres ANSI.

carácter de continuación de línea

La combinación de un espacio seguido de un subrayado (); se utiliza en el entorno de desarrollo para ampliar una sola línea lógica del código hasta dos o más líneas físicas. Observe que los caracteres de continuación de línea no se pueden utilizar para continuar una línea de código dentro de una expresión de tipo cadena.

carácter de declaración de tipo

Un carácter adjunto al nombre de una variable que indica el tipo de datos de una variable. Como opción predeterminada, las variables son del tipo **Variant**, a menos que la instrucción **Deftipo** correspondiente esté presente en el módulo.

clase de base

Clase original de la que otras clases se derivan por herencia.

clase

Definición formal de un objeto. La clase actúa como plantilla desde la que se crea una instancia de un objeto en tiempo de ejecución. La clase define las propiedades del objeto y los métodos utilizados para controlar su comportamiento.

colección

Objeto que contiene un conjunto de objetos relacionados. La posición de un objeto en la colección puede cambiar cada vez que se produzca un cambio en la colección; por lo tanto, la posición de un objeto específico de la colección puede variar.

comentario

Texto agregado a un código por un programador, que explica cómo funciona el código. En Visual Basic cada línea de comentario comienza con un apóstrofo (') o con la palabra clave **Rem** seguida por un espacio.

comparación bit a bit

Comparación bit a bit de los bits ubicados en posiciones idénticas en dos expresiones numéricas.

comparación de cadenas

Una comparación de dos secuencias de caracteres. Utilice **Option Compare** para especificar comparación de texto o binaria. En español, las comparaciones binarias reconocen mayúsculas y minúsculas; las comparaciones de texto no reconocen mayúsculas ni minúsculas.

complemento

Herramienta que agrega capacidades al entorno de programación de Visual Basic.

configuración regional

Conjunto de información que corresponde a un idioma y país determinado. Los valores de la configuración regional afectan al idioma de los términos como palabras clave y define los valores de la configuración regional específica como los separadores decimales y de listas, formatos de fecha y ordenación de caracteres.

Los valores de la configuración regional del sistema afectan al comportamiento del funcionamiento de la configuración regional independiente, por ejemplo, cuando muestra números o convierte cadenas a fechas. Establezca la configuración regional del sistema utilizando los programas del **Panel de Control** que proporciona el sistema operativo.

Aunque la configuración regional del código y del sistema generalmente tienen el mismo valor, puede ser diferentes en algunas situaciones. Por ejemplo, en Visual Basic edición estándar y Visual Basic edición profesional, el código no se traduce del Inglés (Estados Unidos). La configuración regional del sistema se puede establecer al idioma y el país del usuario, pero la configuración regional del código se establece siempre a Inglés (Estados Unidos) y no se puede cambiar. En este caso se utilizan los separadores del Inglés (Estados Unidos), marcadores de posición de formatos y ordenación.

constante de cadena

Cualquier constante (definida con la palabra clave **Const**) que consista en una secuencia de caracteres contiguos interpretados como caracteres por sí mismos, en vez de como algún valor numérico.

constante de compilación condicional

Un identificador de Visual Basic que se define utilizando la directiva de compilador **#Const**, o bien en la aplicación principal y se utiliza en otras directivas de compilador para determinar si determinados bloques de código Visual Basic están compilados, y cuándo.

constante

Un elemento con nombre que mantiene un valor constante a través de la ejecución de un programa. Las constantes pueden ser cadenas o literales numéricos, otras constantes o cualquier combinación que incluya operadores aritméticos o lógicos exceptuando **Is** y la exponenciación. Cada aplicación principal puede definir su propio conjunto de constantes. El usuario puede definir constantes adicionales con la instrucción **Const**. Puede utilizar constantes en cualquier lugar del código en lugar de valores reales.

constantes intrínsecas

Constante proporcionada por una aplicación. Las constantes de Visual Basic se enumeran en la biblioteca de objetos y se pueden mostrar con el **Examinador de objetos**. Ya que no puede deshabilitar constantes intrínsecas, no puede crear una constante definida por el usuario con el mismo nombre.

contenedor

Objeto que puede contener otros objetos.

control ActiveX

Objeto que coloca en un formulario para habilitar o deshabilitar la interacción del usuario con una aplicación. Los controles ActiveX tienen eventos que se pueden incorporar en otros controles. Estos controles tienen una extensión de nombre de archivo .ocx.

control dependiente

Control de datos protegidos que puede proporcionar acceso a un campo o campos específicos en una base de datos mediante un control **Data**.

Normalmente un control de datos protegido depende de un control **Data** mediante sus propiedades **DataSource** y **DataField**. Cuando un control **Data** se mueve de un registro al siguiente, todos los controles dependientes conectados al control **Data** cambian para mostrar los datos de los campos del registro activo. Cuando los usuarios cambian los datos en un control dependiente y después se mueven a un registro diferente, los cambios se guardan automáticamente en la base de datos.

control

Objeto que puede colocar en un formulario que tiene su propio conjunto de propiedades, métodos y eventos reconocidos. Puede utilizar controles para recibir entradas del usuario, mostrar resultados y desencadenar procedimientos de evento. Puede manipular la mayoría de los controles utilizando métodos. Algunos controles son interactivos (responden a acciones del usuario) mientras que otros son estáticos (sólo son accesibles mediante código).

cuadro Objeto

Cuadro de lista que aparece en la esquina superior izquierda de la ventana
Código que enumera el formulario y los controles de formulario al que se adjunta el código o un cuadro de lista colocado en la parte superior de la ventana
Propiedades que enumera el formulario y sus controles.

cuadro Procedimiento

Cuadro de lista colocado en la esquina superior derecha de la ventana **Código** y la ventana **Depuración** que muestra los procedimientos reconocidos por el objeto mostrado en el cuadro **Objeto**.

DBCS

Juego de caracteres que utiliza 1 o 2 bytes para representar un carácter, que permite representar más de 256.

declaración

Código no ejecutable que asigna nombre a una constante, variable o procedimiento y especifica sus características (como el tipo de datos). Para procedimientos DLL, las declaraciones especifican nombres, bibliotecas y argumentos.

directivas de compilación

Comando utilizado para modificar la acción del compilador.

diseñador

Proporciona un generador de ventana visual en el entorno de programación de Visual Basic. Puede utilizar esta ventana para generar visualmente nuevas clases. Visual Basic generadores de formularios incorporados. Las ediciones Profesional y Empresarial de Visual Basic incluyen generadores de controles ActiveX y documentos ActiveX.

documento

Cualquier trabajo con un contenido autónomo creado con una aplicación y se especifica con un nombre de archivo único.

Empty

Un valor que indica que una variable **Variant** no tiene asignado ningún valor inicial. Las variables **Empty** son iguales a 0 (en un contexto numérico) o de longitud cero ("") (en un contexto de cadena).

en proceso

Ejecución en el mismo espacio de direcciones como una aplicación.

enfoque

La capacidad para recibir clics del *mouse* (ratón) o entradas por teclado en cualquier momento. En el entorno de Microsoft Windows, solamente una ventana, un formulario o un control tienen esta capacidad simultáneamente. El objeto que "tiene el foco" se indica normalmente por un título o una barra de títulos resaltados. Se puede establecer el foco tanto por el usuario como por la aplicación.

entorno de programación

Parte de la aplicación donde escribe el código, crea controles, establece las propiedades del control y el formulario, etc... Esto es lo contrario a la ejecución de la aplicación.

error de sintaxis

Un error que ocurre cuando el sistema no comprende una línea de código que acaba de ser introducida.

Observe que las reglas de sintaxis para palabras clave individuales están definidas en la sección de sintaxis del tema de la Ayuda asociado. Para obtener ayuda sobre una palabra clave desde el entorno de desarrollo, seleccione la palabra clave y presione F1.

error en tiempo de ejecución

Un error que ocurre cuando el código se está ejecutando. Un error de ejecución aparece cuando una instrucción intenta llevar a cabo una operación no válida.

error lógico

Un error de programación que hace que el código produzca resultados incorrectos o que detenga la ejecución. Por ejemplo, un error puede ser causado por nombres de variables incorrectos, tipos de variables incorrectos, bucles infinitos, fallos en las comparaciones lógicas o problemas de matrices.

etiqueta de línea

Una etiqueta de línea se usa para identificar una sola línea de código. Puede ser cualquier combinación de caracteres que comience con una letra y que termine con dos puntos (:). Las etiquetas de línea no distinguen mayúsculas y minúsculas y deben empezar en la primera columna.

Examinador de objetos

Un cuadro de diálogo que le permite examinar el contenido de una biblioteca de objetos para obtener información sobre los objetos que proporciona.

expresión Boolean

Expresión cuyos valores son **True** o **False**.

expresión de cadena

Cualquier expresión cuyo valor es equivalente a una secuencia de caracteres contiguos. Los elementos de la expresión pueden incluir una función que devuelve una cadena, un literal de cadena, una constante de cadena, una variable de cadena, una cadena **Variant** o una función que devuelve una cadena **Variant** (**VarType 8**).

expresión de fecha

Toda expresión que se puede interpretar como una fecha. Esto incluye cualquier combinación de literales de fechas, números que parecen fechas, cadenas que parecen fechas y fechas devueltas por funciones. Una expresión de fecha se limita a números o cadenas, en cualquier combinación, que pueda representar una fecha desde el 1 de enero del año 100 hasta el 31 de diciembre del año 9999.

Las fechas se guardan como parte de un número real. Los valores a la izquierda del decimal representan la fecha; los valores a la derecha del decimal representan la hora. Los números negativos representan las fechas anteriores al 30 de diciembre de 1899.

expresión de hora

Cualquier expresión que puede ser interpretada como una hora. Esto incluye cualquier combinación de literales de hora, números que parecen horas, cadenas que parecen horas y horas devueltas por funciones.

Las horas se almacenan como parte de un número real. Los valores a la derecha del decimal representan la hora. El mediodía (12:00 P.M.) se representa con 0,5.

expresión de inspección

Expresión definida por el usuario que le permite observar el comportamiento de una variable o expresión. Las expresiones de inspección aparecen en la ventana **Inspección** del **Editor de Visual Basic** y se actualiza automáticamente cuando entra en el modo de interrupción. La ventana **Inspección** muestra el valor de una expresión dentro de un contexto determinado. Las expresiones de inspección no se guardan con el código.

expresión de objeto

Una expresión que especifica un objeto particular. Esta expresión puede incluir cualquier elementos del objeto. Por ejemplo, si la aplicación tiene un objeto **Aplicación** que contiene un objeto **Documento**, el cual a su vez contiene un objeto **Texto**.

expresión de tipo variant

Cualquier expresión que se pueda evaluar a datos numéricos, de cadena o de fecha, así como a los valores especiales **Empty** y **Null**.

expresión numérica

Cualquier expresión que puede ser evaluada como un número. Los elementos de una expresión pueden incluir cualquier combinación de palabras clave, variables, constantes y operadores que dan como resultado un número.

expresión

Una combinación de palabras clave, operadores, variables y constantes, que produce una cadena, un número o un objeto. Una expresión puede realizar un cálculo, manipular caracteres o verificar datos.

formato automático

Característica que da formato a códigos automáticamente, a medida que se introducen, colocando mayúsculas en la primera letra de las palabras clave, igualando los espacios, agregando la puntuación y configurando los colores del primer plano y del fondo.

formato de fecha universal

El formato de fecha universal es #aaaa-mm-dd hh:mm:ss#. Sin embargo, tanto el componente de fecha (#aaaa-mm-dd#) como el componente de hora (#hh:mm:ss#) se pueden representar por separado.

formulario MDI

Ventana que compone el segundo plano de una aplicación de interfaz de múltiples documentos (MDI). El formulario MDI es el contenedor de cualquiera de los formularios secundarios MDI en la aplicación.

formulario

Ventana o cuadro de diálogo. Los formularios son contenedores para controles. Una formulario de interfaz de múltiples documentos (MDI) también puede actuar como un contenedor para formularios secundarios y algunos controles.

icono

Representación gráfica de un objeto o un concepto, que se utiliza normalmente para representar aplicaciones minimizadas en Microsoft Windows. Un icono es un mapa de bits con un tamaño máximo de 32 x 32 píxeles. Los iconos tienen una extensión de nombre de archivo .ico.

identificador

Elemento de una expresión que hace referencia a una constante o una variable.

indicador de margen

Icono mostrado en la barra del **Indicador de margen** en la ventana **Código**. Los indicadores de margen proporcionan guías visuales durante la modificación del código.

instrucción

Una unidad sintácticamente completa que expresa un tipo de acción, declaración o definición. Normalmente una instrucción tiene una sola línea aunque es posible utilizar dos puntos (:) para poner más de una instrucción en una línea. También se puede utilizar un carácter de continuación de línea (\) para continuar una sola línea lógica en una segunda línea física.

intercambio dinámico de datos (DDE)

Un protocolo establecido para intercambiar datos a través de vínculos activos entre aplicaciones que se ejecutan bajo Microsoft Windows.

juego de caracteres ANSI

Juego de caracteres de 8 bits del Instituto Americano de Normas Nacionales (ANSI), usado por Microsoft Windows, que permite representar hasta 256 caracteres (0–255) con el teclado. Los primeros 128 caracteres (0–127) corresponden a las letras y símbolos de un teclado estándar de EE.UU. Los siguientes 128 caracteres (128–255) representan caracteres especiales, como letras de otros alfabetos, acentos, símbolos de moneda y fracciones.

juego de caracteres ASCII

Juego de caracteres de 7 bits denominado ASCII (Código Americano Estándar para Intercambio de Información), ampliamente utilizado para representar letras y símbolos de en un teclado estándar de EE.UU. El juego de caracteres ASCII es igual que los primeros 128 caracteres (0–127) del juego de caracteres ANSI.

línea de comandos

Información de ruta, nombre de archivo y argumento proporcionada por el usuario para ejecutar un programa.

literal de cadena

Cualquier expresión que consiste en una secuencia de caracteres contiguos rodeados de comillas y que se interpreta literalmente como los caracteres encerrados entre comillas.

literal de fechas

Toda secuencia de caracteres con un formato válido ubicada dentro de símbolos de número(#). Formatos válidos incluyen el formato de fecha especificado por la configuración correspondiente a la localidad del código o al formato de fecha universal.

Por ejemplo, #31/12/92# es el literal de fecha que representa el 31 de diciembre de 1992, en este caso, Español-España es la configuración correspondiente a la localidad de la aplicación. El uso de literales de fecha maximiza la transportabilidad a través de los diferentes lenguajes nacionales.

llamada a procedimiento

Instrucción en código que indica a Visual Basic que ejecute un procedimiento.

método gráfico

Método que opera en objetos como **Form**, **PictureBox** o **Printer** y ejecuta operaciones de dibujo en tiempo de ejecución como animación o simulación. Los métodos gráficos son **Circle**, **Cls**, **Line**, **PaintPicture**, **Point**, **Print** y **PSet**.

método

Un procedimiento que se aplica a un objeto.

módulo de código

Módulo que contiene código público que se puede compartir entre todos los módulos de un proyecto. Los módulos de código se denominan módulos estándar en versiones posteriores de Visual Basic.

módulo de clase

Módulo que contiene la definición de una clase (sus definiciones de propiedad y método).

módulo de formulario

Archivo en un proyecto de Visual Basic con una extensión de nombre de archivo .frm que puede contener descripciones gráficas de un formulario; sus controles y sus valores de propiedad; declaraciones a nivel de formulario de constantes, variables y procedimientos externos; eventos y procedimientos generales.

módulo de objeto

Módulo que contiene un código específico a un objeto, por ejemplo, módulo de clase, módulo de formulario y módulo de documento. Los módulos objeto contienen el código detrás de sus objetos asociados. Las reglas para módulos objeto difieren de las de los módulos estándar.

módulo estándar

Un módulo que contiene solamente declaraciones y definiciones de procedimiento, tipo y datos. Las declaraciones y definiciones a nivel de módulo de un módulo estándar son **Public** de manera predeterminada. Un módulo estándar se denomina módulo de código en versiones anteriores de Visual Basic.

módulo

Un conjunto de declaraciones y procedimientos.

mapa de bits

Imagen representada por píxeles y almacenada como una colección de bits en la que cada bit corresponde a un píxel. En equipos con pantalla en color, a cada píxel le corresponde más de un bit. Normalmente un mapa de bits tiene una extensión de nombre de archivo .bmp.

marco de ventana vinculada

Marco de ventana que contiene múltiples ventanas que se han vinculado juntas.

matriz de controles

Grupo de controles que comparten un nombre, tipo y procedimientos de evento comunes. Cada control en una matriz tiene un número de índice único que se puede utilizar para determinar qué control reconoce un evento.

matriz

Conjunto de elementos que tienen el mismo tipo de datos y que están ordenados secuencialmente. Cada elemento de una matriz posee un número de índice único que lo identifica. Los cambios efectuados a un elemento de una matriz no afectan a los demás elementos.

MDI secundario

Formulario contenido dentro de un formulario MDI en una aplicación de interfaz de múltiples documentos (MDI). Para crear un formulario secundario, establezca la propiedad **MDIChild** del formulario MDI a **True**.

metarchivo

Archivo que almacena una imagen como objetos gráficos tanto líneas, círculos y polígonos como píxeles. Hay dos tipos de metarchivos, estándar y ampliado. Los metarchivos estándar normalmente tiene una extensión de nombre de archivo .wmf. Los metarchivos ampliados normalmente tiene una extensión de nombre de archivo .emf. Los metarchivos preservan una imagen más exacta que los píxeles cuando la imagen se cambia de tamaño.

miembro

Los elementos que constituyen un conjunto, un objeto o un tipo definido por el usuario.

modo de interrupción

Suspensión temporal de la ejecución de un programa mientras está en la etapa de desarrollo. En el modo interrumpir, se puede examinar, depurar, volver a configurar, incrementar o continuar una ejecución. Se entra en modo interrumpir:

Al encontrar un punto de interrupción durante la ejecución del programa.

Al presionar Ctrl+Pausa durante la ejecución del programa.

Al encontrar una instrucción **Stop** o un error de ejecución no detectado durante la ejecución del programa.

Agregando una expresión **Interrumpir cuando el valor sea Verdadero**. La ejecución se detiene cuando el valor de la expresión bajo inspección cambia y se convierte en **True**.

Agregando una expresión **Interrumpir cuando el valor cambie**. La ejecución se detiene cuando el valor de la expresión bajo inspección cambie.

número de archivo

Número utilizado en la instrucción **Open** para abrir un archivo. Utilice los números de archivo en el intervalo 1–255 inclusive, para archivos no accesibles desde otras aplicaciones. Utilice números de archivo en el intervalo 256–511 inclusive, para archivos accesibles desde otras aplicaciones.

número de error

Un número completo en el intervalo que comprende de 0 a 65.535, inclusive, que se corresponde con la propiedad **Number** del objeto **Err**. Cuando se combina con la propiedad **Description** del objeto **Err**, este número representa un mensaje de error determinado.

número de línea

Un número de línea se usa para identificar una sola línea de código. Este número puede ser cualquier combinación de dígitos que sea única dentro del módulo donde se usa. Los números de línea deben comenzar en la primera columna.

nivel de módulo

Código en la sección de declaraciones de un módulo. Cualquier código fuera de un procedimiento se denomina código de nivel de módulo. Las declaraciones se deben colocar primero, seguidas de los procedimientos.

nivel de procedimiento

Instrucciones localizadas dentro de los procedimientos **Function**, **Property** o **Sub**. Generalmente, las declaraciones aparecen primero, seguidas de asignaciones y otro código ejecutable.

Observe que el código de nivel de módulo reside fuera del bloque de procedimiento.

Null

Un valor que indica que una variable contiene datos no válidos. **Null** es el resultado de una asignación explícita de una variable como **Null** o cualquier operación entre expresiones que contienen **Null**.

objeto ActiveX

Objeto que se expone a otras aplicaciones o herramientas de programación mediante interfaces de Automatización.

objeto de Automatización

Objeto que se ofrece para otras aplicaciones o herramientas de programación a través de las interfaces de Automatización.

objeto insertable

Objeto de aplicación que es un tipo de control personalizado, como una hoja de cálculo de Microsoft Excel.

objeto origen de evento

Objeto que es el origen de eventos que se producen en respuesta a una acción. Una propiedad devuelve un origen de evento. Por ejemplo, la propiedad **CommandBarEvents** devuelve el objeto **CommandBarEvents**.

objeto

Combinación de código y datos que se pueden tratar como una unidad, por ejemplo, un control, un formulario o un componente de una aplicación. Cada objeto se define por una clase.

operador de comparación

Símbolo o un carácter que indica la relación entre dos o más valores, o expresiones. Estos operadores incluyen menor que (<), menor o igual que (<=), mayor que (>), mayor o igual que (>=), distinto (<>) e igual (=). Los operadores de comparación adicionales incluyen **Is** y **Like**. Observe, sin embargo, que **Is** y **Like** no se pueden utilizar como operadores de comparación en la instrucción **Select Case**.

orden de tabulación

Orden en el que el enfoque se mueve de un campo al siguiente cuando presiona Tab o Mayús+Tab.

orden z

Disposición visual de controles en un formulario a lo largo del eje z del formulario (profundidad). El orden z determina qué controles están por delante de los otros.

orden

Un principio de secuencia que se usa para ordenar datos; por ejemplo, alfabético, numérico, ascendente, descendente, etc.

palabra clave

Una palabra o un símbolo reconocido como parte del lenguaje de programación; por ejemplo, una instrucción, un nombre de función o un operador.

panel de código

Panel contenido en una ventana de código que se utiliza para introducir y modificar código. Una ventana de código puede contener uno o más paneles de código.

parámetro

Nombre de variable por el cual un argumento de procedimiento se conoce dentro del procedimiento. Esta variable recibe el argumento que se pasa dentro del procedimiento. Su alcance termina cuando el procedimiento termina.

pi

Pi es una constante matemática aproximadamente igual a 3,1415926535897932.

pila

Cantidad fija de memoria utilizada por Visual Basic para preservar las variables locales y los argumentos durante llamadas a procedimientos.

por referencia

Modo de pasar un argumento a un procedimiento pasando su dirección en vez de su valor. Esto hace posible el acceso del procedimiento a la variable real. Como resultado, el valor real de la variable puede ser cambiado por el procedimiento al cual se pasa.

por valor

Modo de pasar un argumento a un procedimiento pasando su valor en vez de su dirección. Esto hace posible el acceso del procedimiento a una copia de la variable. Como resultado, el valor actual de la variable no puede ser cambiado por el procedimiento al cual se pasa.

Private

Variables que sólo son visibles en el módulo que se declaran.

procedimiento Function

Procedimiento que ejecuta una tarea específica dentro de un programa y devuelve un valor. Un procedimiento **Function** comienza con una instrucción **Function** y finaliza con una instrucción **End Function**.

procedimiento general

Procedimiento que se debe llamar explícitamente desde otro procedimiento. Por el contrario, un procedimiento de evento se invoca automáticamente en respuesta a una acción de usuario o del sistema.

procedimiento Property

Procedimiento que crea y manipula propiedades para un módulo de clase. Un procedimiento **Propiedad** comienza con una instrucción **Property Let**, **Property Get** o **Property Set** y termina con una instrucción **End Property**.

procedimiento Sub

Procedimiento que ejecuta una tarea específica dentro de un programa, pero devuelve un valor no explícito. Un procedimiento **Sub** comienza con una instrucción **Sub** y termina con una instrucción **End Sub**.

procedimiento

Una secuencia con nombre de instrucciones que se ejecutan como una unidad. Por ejemplo, **Function**, **Property** y **Sub** son todos tipos de procedimientos. Un nombre de procedimiento siempre se define a nivel de módulo. Todo el código ejecutable debe estar contenido en un procedimiento. Los procedimientos no se pueden anidar dentro de otros procedimientos.

propiedad

Un atributo con nombre de un objeto. Las propiedades definen características de objetos, como tamaño, color y ubicación en la pantalla, o comportamientos de objetos, como si está o no habilitado.

proyecto al que se hace referencia

El proyecto en el cual se está trabajando se denomina proyecto actual. Se pueden crear vínculos a otros proyectos desde el proyecto actual. El proyecto al que se vincula se denomina proyecto referido directamente. Sus símbolos **Public** no son accesibles al proyecto actual, excepto mediante calificación con el nombre del proyecto. Cualquier combinación de referencias directas e indirectas entre proyectos es válida, siempre y cuando no resulte en un ciclo completo.

proyecto de referencia

El proyecto actual se denomina proyecto de referencia directa. La forma de crear un vínculo con un proyecto depende de la aplicación principal. Por ejemplo, en Excel puede hacer referencia directamente a un proyecto eligiéndolo en el cuadro de diálogo **Referencias** del menú **Herramientas**. Los símbolos **Public** de un proyecto referido directamente son visibles para el proyecto de referencia directa, pero los símbolos **Public** de un proyecto de referencia directa no son visibles para un proyecto referido directamente.

proyecto

Un conjunto de módulos.

Public

Variables declaradas utilizando la instrucción **Public** que son visibles a todos los procedimientos en todos los módulos y en todas las aplicaciones a menos que **Option Private Module** esté activa. En este caso, las variables son públicas sólo dentro del proyecto en el que residen.

punto de interrupción

Una línea seleccionada del programa en la cual la ejecución del programa se detiene automáticamente. Los puntos de interrupción no se guardan con su código.

punto

Un punto es $1/72$ de pulgada. Los tamaños de fuente normalmente se miden en puntos.

registro

En Windows versión 3.1, la información de registro OLE y las asociaciones de archivo se almacenan en la base de datos de registro, y la configuración del programa se almacena en archivos de inicialización del sistema (.INI) de Windows. En Windows 95, el registro de Windows sirve como base de datos central de configuración para información específica del ordenador, de usuario y de aplicación, incluyendo la información que previamente se situaba en la base de datos de registro y en los archivos .ini.

ruta de acceso

Una expresión de cadena que especifica la ubicación de un directorio o carpeta; puede incluir una especificación de unidad de disco.

semilla

Un valor inicial usado para generar números pseudoaleatorios. Por ejemplo, la instrucción **Randomize** crea un número semilla que usa la función **Rnd** para crear secuencias de números pseudoaleatorios únicas.

separadores de fecha

Caracteres utilizados para separar el día, el mes y el año cuando se da formato a valores que representan fechas. Los caracteres están determinados por las configuraciones de sistema o por la función **Format**.

tecla de acceso

Tecla que mientras mantiene presionada la tecla Alt permite al usuario abrir un menú, ejecutar un comando, seleccionar un objeto o moverse a un objeto. Por ejemplo, Alt+A abre el menú **Archivo**.

tiempo de compilación

Momento en el que el código de origen se traduce a código de ejecución.

tiempo de diseño

El tiempo durante el cual se genera una aplicación en el entorno de desarrollo agregando controles, estableciendo propiedades de controles o formularios, etc. Por el contrario, durante el tiempo de ejecución se interactúa con la aplicación como lo haría un usuario.

tiempo de ejecución

El tiempo durante el cual se está ejecutando código. Durante el tiempo de ejecución, se interactúa con el código como lo haría un usuario.

tipo de datos Boolean

Tipo de datos que sólo tiene dos valores posibles, **True** (-1) o **False** (0). Las variables **Boolean** se almacenan como números de 16 bits (2 bytes).

tipo de datos Byte

Tipo de datos utilizado para contener números enteros positivos en el intervalo de 0 a 255. Las variables de tipo Byte se almacenan como Single, números sin signo de 8 bits (1 byte).

tipo de datos Currency (Moneda)

Tipo de datos con un intervalo de -922.337.203.685.477,5808 a 922.337.203.685.477,5807. Utilice este tipo de datos para cálculos de tipo monetario y cálculos de punto fijo donde es especialmente importante la exactitud. El carácter de declaración de tipo es (@) que representa el tipo **Currency** en Visual Basic.

tipo de datos Date (Fecha)

Tipo de datos utilizado para almacenar fechas y horas como un número real. Las variables de tipo Date se almacenan como números de 64 bits (8 bytes). El valor de la izquierda de los decimales representa la fecha y valor de la derecha representa una hora.

tipos de datos decimales

Tipo de datos que contiene números decimales en escala de una potencia de 10. Para números de escala cero, o sea, números sin posiciones decimales, el intervalo es de +/-79.228.162.514.264.337.593.543.950.335. Para números con 28 posiciones decimales el intervalo es +/-7,9228162514264337593543950335. El número más pequeño distinto de cero que se puede representar como un tipo **Decimal** es 0,0000000000000000000000000001.

Observe que en este momento el tipo de datos **Decimal** sólo se puede utilizar dentro de un tipo **Variant**. No puede declarar una variable de tipo **Decimal**. Sin embargo, puede crear un tipo **Variant** cuyo subtipo es **Decimal** utilizando la función **CDec**.

tipo de datos Double (Doble)

Tipo de datos que contiene números de doble precisión de punto flotante como números de 64 bits en el intervalo de $-1.79769313486231E308$ a $-4,94065645841247E-324$ para valores negativos y de $4,94065645841247E-324$ a $1,79769313486232E308$ para valores positivos. El carácter de declaración de tipo es el signo de número (#) que representa el tipo **Double** en Visual Basic.

tipo de datos Integer (Entero)

Tipo de datos que contiene variables enteras almacenadas como números enteros de 2 bytes en el intervalo de -32.768 a 32.767. El tipo de datos **Integer** también se utiliza para representar valores en listas. El carácter de declaración de tipo es el signo de porcentaje (%) que representa un tipo **Integer** en Visual Basic.

tipo de datos Long (Entero largo)

Un entero de cuatro bytes, un número completo entre -2.147.483.648 y 2.147.483.647. El carácter de declaración de tipo es (&) que representa un tipo **Long** en Visual Basic.

tipo de datos numérico

Cualquier tipo de datos numérico intrínseco (**Byte, Boolean, Integer, Long, Currency, Single, Double** o **Date**).

tipo de datos Object

Tipo de datos que representa cualquier referencia a un tipo **Object**. Las variables **Object** se almacenan como direcciones de 32 bits (4 bytes) que hacen referencia a objetos.

tipo de datos Single (Sencillo)

Tipo de datos que almacena variables de precisión sencilla de punto flotante como números de punto flotante de 32 bits (4 bytes) en el intervalo de valores de $-3,402823E38$ a $-1,401298E-45$ para valores negativos y de $1,401298E-45$ a $3,402823E38$ para valores positivos. El carácter de declaración de tipo es el signo de admiración (!) que representa el tipo **Single** en Visual Basic.

tipo de datos String (Texto)

Tipo de datos que consiste en una secuencia de caracteres que representa a los caracteres por sí mismos en vez de sus valores numéricos. Un tipo **String** puede incluir letras, números, espacios en blanco y signos de puntuación. El tipo de datos **String** puede almacenar cadenas de longitud fija en un intervalo de 0 a aproximadamente 63000 caracteres y cadenas dinámicas en un intervalo de longitud de 0 a aproximadamente 2 mil millones de caracteres. El carácter de declaración de tipo es el signo de dólar (\$) que representa el tipo **String** en Visual Basic.

tipo de datos Variant

Un tipo de datos especial que contiene datos numéricos, de cadena o de fecha así como tipos definidos por el usuario y los valores especiales **Empty** y **Null**. El tipo de datos **Variant** tiene un tamaño de almacenamiento numérico de 16 bytes y puede contener datos hasta el intervalo de un tipo **Decimal** o un tamaño de almacenamiento de caracteres de 22 bytes (más la longitud de cadena) y puede almacenar cualquier texto. La función **VarType** define el tratamiento que reciben los datos de un **Variant**. Todas las variables son del tipo **Variant** a menos que se declaren explícitamente como de cualquier otro tipo.

tipo de datos

Característica de una variable que determina qué tipo de datos puede tener. Los tipos de datos incluyen **Byte, Boolean, Integer, Long, Currency, Single, Double, Date, String, Object, Variant** (predeterminado) y tipos definidos por el usuario, así como tipos específicos de objetos.

tipo de objeto

Un tipo de objeto expuesto por una aplicación por medio de la Automatización. Por ejemplo, **Aplicación**, **Archivo**, **Intervalo** y **Hoja de cálculo**. Utilice el **Examinador de objetos** o consulte la documentación de la aplicación para obtener una lista completa de objetos disponibles.

tipo definido por el usuario

Cualquier tipo de datos definido usando la instrucción **Type**. Los tipos de datos definidos por el usuario pueden contener uno o más elementos de cualquier tipo de datos. Las matrices de tipos definidos por el usuario y de otros tipos de datos se crean con la instrucción **Dim**. Se pueden incluir matrices de cualquier tipo dentro de tipos definidos por el usuario.

tipo numérico

Cualquier tipo de datos numérico intrínseco (**Byte, Boolean, Integer, Long, Currency, Single, Double** o **Date**) cualquier subtipo numérico **Variant** (**Empty, Integer, Long, Single, Double, Currency, Decimal, Date, Error, Boolean** o **Byte**).

twip

Unidad de medida de pantalla que es igual a $1/20$ de punto. Un twip es una unidad de pantalla independiente utilizada para asegurar que la ubicación y proporción de los elementos de la pantalla en la aplicación son los mismos en todos los tipos de pantallas. Hay aproximadamente 1440 twips en una pulgada lógica o 567 twips en un centímetro lógico (la longitud de un elemento de pantalla que mide una pulgada o un centímetro cuando se imprime).

Unicode

Carácter estándar de la Organización internacional de estándares (*International Standards Organization*) (ISO). Unicode utiliza un esquema de códigos de 16-bits (2 bytes) que permite 65.536 espacios de caracteres distintos. Unicode incluye las representaciones de signos de puntuación, símbolos matemáticos y dingbats, con amplio espacio para futuras ampliaciones.

variable de módulo

Una variable declarada fuera del código de los procedimientos **Function**, **Sub** o **Property**. Las variables de módulo se deben declarar antes de cualquier procedimiento en el módulo. Existen mientras el módulo está cargado, y son visibles en todos los procedimientos del módulo.

variable de objeto

Variable que contiene una referencia a un objeto.

variable

Un lugar de almacenamiento con nombre que puede contener cierto tipo de datos que puede ser modificado durante la ejecución del programa. Cada variable tiene un nombre único que la identifica dentro de su nivel de ámbito. Puede especificar un tipo de datos o no.

Nombres de variable deben comenzar con un carácter alfabético, deben ser únicos dentro del mismo ámbito, no deben contener más de 255 caracteres y no pueden contener un punto o carácter de declaración de tipo.

ventana acoplada

Ventana que se adjunta al marco de la ventana principal.

ventana Propiedades

Ventana que se utiliza para mostrar o cambiar las propiedades de un formulario seleccionado en tiempo de diseño. Algunos controles personalizados tienen ventanas **Propiedades** personalizadas.

ventana Proyecto

Ventana que muestra una lista del formulario, la clase y los módulos estándar; el archivo de recursos y las referencias en el proyecto. Los archivos con extensiones de nombre de archivo .ocx y .vbx no se muestran en la ventana **Proyecto**.

ventana vinculada

Ventana que se combina con otra ventana distinta de la ventana principal.

verificación de sintaxis

Es una característica que verifica el código para asegurarse de que la sintaxis es correcta. Si la verificación de sintaxis está activada, se mostrará un mensaje cuando se introduzca código que tenga un error de sintaxis y la parte de código correspondiente aparecerá resaltada.

zona de impresión

Las zonas de impresión comienzan cada 14 columnas. El ancho de cada columna es un promedio del ancho de todos los caracteres, medido con el tamaño de punto de la fuente elegida.