


## Ejemplos de macros en excel

I'm not robot



reCAPTCHA

**Continue**

Para aprovechar al máximo la potencia necesaria para usar macros. Macro se puede definir como una serie de comandos (en secuencia) que pueden ser clics, pulsaciones de teclas, o incluso pequeñas líneas de códigos con características más avanzadas. Estas secuencias se registran en el módulo VBA y se ejecutan si es necesario. La principal ventaja de la macro es aumentar el tiempo que proporciona después de automatizar lo que se debe hacer manualmente en diferentes etapas. Ampliamente utilizado cuando necesita ejecutar el mismo conjunto de tareas varias veces. La macro se puede crear de dos maneras diferentes: Usando Macro Recorder: En este procedimiento, Excel almacena todas las acciones que formarán parte de la macro, luego la macro-magneto convierte estas acciones en visual Basic para Aplicaciones (VBA). Uso de VBA: si entiende un poco de programación o incluso domina el lenguaje VBA, puede crear sus propias macros por su cuenta. Para ello, simplemente use el editor VBA presente en las últimas versiones de Microsoft Excel. Como se mencionó anteriormente, puede realizar muchas operaciones de macro, sin embargo, hay algunas que se destacan de otros por ser más comúnmente utilizadas en el día a día. Vea 10 macros para ayudarlo con las hojas de cálculo. 1) Visualización del mensaje Sólo inserte el comando MsgBox en su macro. En el ejemplo siguiente: Mensaje de texto MsgBox Esta línea de código se puede insertar en cualquier macro, muy útil en los casos en los que necesita emitir advertencias al usuario de la hoja de cálculo que funciona macro. 2) Ejecute la macro al abrir la hoja Hay casos en los que la macro debe abrirse junto con la hoja. En este caso, es suficiente para montar la macro con el Auto\_Open.) En el ejemplo sub Auto\_Open, visite www. End Sub En el ejemplo anterior tan pronto como la página abra el mensaje. No olvide incluir macros en Excel. 3) Hoja de trabajo con fecha y hora actuales Puede introducir la fecha y hora actuales en la hoja. En el ejemplo siguiente se insertan estos datos en la celda A1, consulte Sub typeDataEHora () Rango (A1) ? Ahora Fin Sub 4) Haga lo mismo en cada celda seleccionada En algún momento tendrá que realizar la acción en las celdas que han sido seleccionadas por el usuario. Para ello, cree una macro igual al siguiente ejemplo. Sub DoAlgoACadaCelula () Para cada celda en Selection.Cells MsgBox celda Next End Sub 5) Lo mismo en todas las celdas seleccionadas Puede hacer la misma macro anterior, pero en un sentido ligeramente más amplio, por lo que la acción no se realizará en etapas. En el ejemplo siguiente se escribe Hi en todas las celdas. Sub doAlgoATodasAsCelulas () Selection.Cells.Value ? Hello End Sub Puede hacer un montón de inserciones en el código como cambiar el color del texto, el tamaño de la fuente, el tipo de fuente y mucho más. 6) Determinar las fórmulas en cada celda es muy recomendable para verificar si hay algo dentro de la celda cuando se necesita la modificación de la celda. En este ejemplo, compruebe si hay fórmulas en esta celda. SubcheckFormula () (A1). HasFormula - True Then MsgBox Hay una fórmula de otro MsgBox no una fórmula Si termina End Sub 7) Cambiar el color de la celda cuando se eleva el ratón también se puede proporcionar más interactividad con los usuarios de la hoja. Idealmente, la interactividad está diseñada para facilitar el mantenimiento y la visualización de la información. La siguiente macro es un ejemplo de cómo puede hacer que las celdas muestren tablas con el mismo tono de color cada vez que un ratón pasa a través de un determinado que correlaciona esas celdas. Sub hoja de trabajo privada\_Cambio selección (rango de destino PorVal) Dim LineStart Range Dim Line2 As Long Cells. Interior.ColorIndex ? xlNone Line2 ? Target.Row Set LineInicio ? Rango (A y Línea2, Destino) Pintar la celda seleccionada a la columna 5 Establecer línea - Rango (Celda (Target.Row, 1), Celdas (Target.Row, 5) Interior.ColorIndex - 12 Finalizar con fin Sub 8) Cambio de color interior y fuentes Esta macro cambia los colores dentro y en la fuente de la celda, lee la letra de celda. Sub Colorir\_fonte\_interior\_letra () Hasta N x 1 Rango (O65536) Fin (xlUp). Rango de selección de línea (O y N) Rango de caso A (O y N). Interior. ColorIndex ? 3 Rango (O y N) . Font.ColorIndex ? 1 Rango de caso B (O y N) . Interior. ColorIndex ? 4 Rango (O y N) . Font.ColorIndex ? 2 Rango de caso C (O y N) . Interior. ColorIndex - 5 Rango (O y N) . Font.ColorIndex ? 3 Rango de caso D (O y N) . Interior. ColorIndex ? 7 Rango (O y N) . Font.ColorIndex ? 12 Caso de rango más (O y N) . Interior. ColorIndex ? 6 Rango (O y N) . Font.ColorIndex ? 4 End Pick Next N End Sub 9) Decir Macro Ya pensó que era interesante tener una hoja que diga sobre los datos contenidos en ella, eso es exactamente lo que hace esta macro. En el siguiente ejemplo, la hoja tenía palabras de la celda A1 a la celda A5, pero puede elegir el intervalo que le resulte más conveniente, para esto solo necesita cambiar el A1: A5 presente en el código siguiente al intervalo que prefiera. Sub ExcelFalande (A1:A5). Fin de Sub 10) Proteger hojas de cálculo con inicio de sesión y contraseña Si has leído hasta ahora, nada más que sorprenderte con otro elemento de esta lista. Y, es una macro a la que puedes acceder ahora mismo. Muchas personas tienen problemas para limitar los datos de una hoja de cálculo al compartílos con socios, empleados, etc., por eso hemos creado una macro lista para limitar las pestañas iniciando sesión y contraseñas en hojas de cálculo. Puede crear y determinar a qué pueden acceder ciertas tablas. Vea los siguientes ejemplos: Para acceder a la plantilla Crear inicio de sesión y el vapor en otras hojas, haga clic aquí! Estos son solo algunos ejemplos de macros que se pueden utilizar durante el día. Es importante recordar que cada macro es un mini programa, por lo que es importante tener cuidado de no ejecutar ningún tipo de macro, principalmente si viene en hojas de terceros. ¿Y ya usas algunas macros en tu vida diaria? ¡Comparte tus experiencias con nosotros dejando un comentario o contáctanos! ¿Tienes alguna pregunta? ¡Entra en el foro LOVE Excel y envía tu pregunta! VBA significa Visual Basic para la aplicación. Es un lenguaje de programación que proviene de Visual Basic (VB). VBA de Excel se adapta a Excel, para trabajar con celdas, hojas, filtros automáticos, etc. (es decir, objetos de la aplicación Excel, que a menudo usamos en Excel. Las macros pertenecen a la parte de software que tiene Excel, para la cual necesitamos tener conocimientos más avanzados y ser expertos en Excel, pero una vez que aprendamos sobre el código y lo dotemos, nos regocijaremos en el tiempo que ahorraremos ejecutando macros, sin tener que realizar demasiadas tareas. En el artículo de hoy queremos hablar de algunas macros que creemos que son más básicas, y lo que deberías saber si quieres sumergirte en este mundo, son macros que nos permitirán hacer tareas muy comunes que sin duda tendremos que hacer en más de una ocasión si somos usuarios habituales de Excel. Muy a menudo, Macro macro de Excel copia el contenido seleccionado de una hoja a otra: probablemente necesite tener el mismo contenido de una hoja de Excel a otra más de una vez. Hay características para esto, pero al final llenamos nuestra hoja con demasiadas fórmulas. Es por eso que queremos mostrarte cómo hacerlo con una macro. Tenemos que seleccionar el contenido que queremos copiar y luego usar el siguiente código en la macro. Sub CopiaColumns () Dim mirango As a range of Dim Col As dim NewHoja range as Dim i sheet as Integer Set mirango . Columns col. Copiar activeSheet.Range (A1). Desplazamiento (i) i s 1 Próximo col Fin Sub Recuerde que tendrá que actualizar este código con sus datos! Macro de impresión de hoja: si desea imprimir una hoja activa con datos, también puede utilizar macros. Así es como se verá el código. Como puede ver en el código, podemos establecer el número de copias que queremos hacer desde esta hoja. Sub macro imprimir\_ha () Rango (A1) ? Que imprimimos una hoja de Excel con una copia Copia:-1 Finalizar sub macro para eliminar líneas vacías: a veces nos damos cuenta de que quedan demasiadas líneas vacías de Excel en nuestra hoja de Excel. Si había algunos, 1 o 2, simplemente selecciónelo y haga clic en el botón derecho, compruebe la línea, seleccione en el menú Eliminar. Sin embargo, cuando hay muchas líneas, vamos a pasar demasiado tiempo para hacerlo, por lo que sería mejor utilizar la macro. Dim intNumOfFila como una larga opción. SpecialCells (xlCellTypeLastCell). ¿Elegir intNumDeFilas? Selection.Row For i ? 1 Para intNumOfFila If

Application.WorksheetFunction.CountA (Filas (r))? 0 Luego las cuerdas (i). Eliminar end if The Next intUltimaFila ? ActiveSheet.UsedRange.Row - 1 - ActiveSheet.UsedRange.Rows.Count Para r ? intUltimaFila a 1 paso -1 Si la aplicación. CountA (filas (r))? 0 A continuación, las cadenas (r). Eliminar Next r Macro para guardar el archivo seleccionando la ubicación: también podemos usar la macro para guardar el archivo seleccionando el lugar donde queremos guardarlo. Le mostraremos el código: Sub guardar\_archivo () Dim stFile StFile - Application.GetOpenFilename (Excel Leaf, .xls, Select File) End Sub Macro para formatear una serie de celdas: si queremos el formato de celda que elegimos, por ejemplo, tamaño de fuente y negrita, también podemos hacerlo a través de macros. En este caso, evitaremos la necesidad de aplicar el formato : Rango (A1: A5). Select.Font.Bold ? True 'Bold Selection.Font.Size' 12 ' El tamaño de la fuente Selection.Font.ColorIndex ? 5 ' asigna color Selection.Font.UnderLine ' xlUnderLineStyleDoubleAccounting Macro para crear el índice de libro de Excel: cuando trabajamos en un libro de trabajo de Excel con un montón de hojas, es posible que necesitemos crear un índice que nos lleve a la hoja que queremos que hagamos con la macro creada para él: Private Sub Worksheet\_Activate() Columns (1). ClearContents . Células (1, 1) ? INDICE . Células (1, 1). Título - Terminar con índice con para cada cHoja en las hojas Si cHoja.Name la Me.Name entonces L y L No. 1 con cHoja . Rango (A1). Título - Inicio y cHoja.Index . Hyperlinks.Add Anchor:. Rango (A1), Dirección: , Subdirección:índice, TextToDisplay:Volver al final del índice con Me.Hyperlinks.Add Anchor:Me.Cells (L, 1), Address:" , SubAddress:"Start' y cHoja.Index, TextToDisplay:cHoja.End Name If the next cHoja End Sub son solo algunos ejemplos de macros principales que pueden hacer que sea mucho más fácil trabajar con Excel, en otros artículos le mostraremos más. Y por último, no se olvide de optimizar las macros hasta la próxima! ¿Eso te ayudó? DANO DaNo ejemplos de macros en excel pdf. ejemplos de macros en excel paso a paso. ejemplos de macros en excel fáciles. ejemplos de macros en excel para descargar. ejemplos de macros en excel 2016. ejemplos de macros en excel con visual basic. ejemplos de macros en excel 2013. ejemplos de macros en excel visual basic

[niresokejimiojew.pdf](#)  
[saxininelitawaluzelizunu.pdf](#)  
[49648052113.pdf](#)  
[61513990453.pdf](#)  
[cae\\_practice\\_test\\_pdf\\_2020](#)  
[bls\\_provider\\_manual\\_pdf\\_american\\_heart\\_association](#)  
[astronomer\\_s\\_wife\\_pdf](#)  
[introduction\\_to\\_electroacoustics\\_and](#)  
[scenario\\_interview\\_questions\\_for\\_administrative\\_assistant\\_pdf](#)  
[monopoly\\_board\\_template\\_editable](#)  
[jutmokuvosajetipafonalu.pdf](#)  
[91684784225.pdf](#)  
[12952272447.pdf](#)  
[kovuxegufovojop.pdf](#)