

wComboBox

wComboBox nos permite incluir en nuestras web un control Select de Materialize <https://materializecss.com/select.html>.

Con el podemos darle al usuario la opción de seleccionar un item dentro de una lista de items que le ofrecemos.

Los items los alimentamos mediante la propiedad `altems` que puede ser utilizada de tres maneras.

1. con un array de elementos (cadenas y/o numeros) {"Elemento1","Elemento2","ElementoX"}, en este caso nos devuelve el elemento seleccionado ("ElementoX").
2. con un array de parejas {Indice,Valor} ({ {cIndex1,cText1},{cIndex2,cText2},{cIndexX,cTextX} }), en este caso devuelve el cIndex del elemento seleccionado (cIndexX). En este caso cIndex con valor -1 serán utilizadas como Grupos y no se pueden seleccionar.
3. con un array de [wItemComboBox](#), se utiliza para poder añadir imágenes y grupos a los ComboBox.

Hay que recordar que el [CGI](#) siempre nos devuelve cadenas, sean los Elementos y cIndex del tipo que sean.

Si habilitamos la propiedad **IMultiple** nos permitirá seleccionar varias opciones de las presentadas en el combo, en ese caso el [CGI](#) nos devolverá un array con los los elementos seleccionados.

Ejemplo de uso

Modo 1

Nos retornara el nombre del mes seleccionado

```
WITH OBJECT WComboBox():New(:W0)
  :cId      := "mesvisita1"
  :aItems   := {"Enero","Febrero","Marzo","Abril","Mayo","Junio",;
"Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"}
  :cText    := "Seleccione el mes de visita"
  :cIcon    := "event_available"
  :cTitle   := "<b>Modo 1:</b> Mes de visita"
  :cHelp    := "Este es el Mes que prefiere que le hagamos la visita
periódica"
  :Create()
END WITH
```

Y para recuperar la selección ...

```
LOCAL cSeleccion := oCGI:GetCgiValue("mesvisita1")
```

Modo 2

Nos presentara los meses separados por trimestres y retornara el número del mes seleccionado

```
WITH OBJECT WComboBox():New(:W0)
  :cId      := "mesvisita2"
  :aItems   := {{-1, "1er
Trimestre"},{1,"Enero"},{2,"Febrero"},{3,"Marzo"},;
              {-1, "2o Trimestre"}, {4,"Abril"},{5,"Mayo"},{6,"Junio"},;
              {-1, "3er Trimestre"},
{7,"Julio"},{8,"Agosto"},{7,"Septiembre"},;
              {-1, "4o Trimestre"},
{10,"Octubre"},{11,"Noviembre"},{12,"Diciembre"}}
  :cText    := "Seleccione el mes de visita"
  :cIcon    := "event_available"
  :cTitle   := "<b>Modo 2:</b> Mes de visita"
  :cHelp    := "Este es el Mes que prefiere que le hagamos la visita
periódica"
  :Create()
END WITH
```

Y para recuperar la selección ...

```
LOCAL cSeleccion := oCGI:GetCgiValue("mesvisita2")
```

Modo 3

Utilizando AddItem podemos añadir Imágenes a nuestro wComboBox

```
WITH OBJECT WComboBox():New(:W0)
  :cId      := "selimagen"
  :cText    := "Seleccione una imagen"
  :cIcon    := "camera"
  :cTitle   := "<b>Modo 3:</b> Imagen para incluir en el documento"
  :cHelp    := "Esta Imagen aparecerá en la parte superior del documento"
  // Creamos un separador
  WITH OBJECT :AddItem()
    :cIndex := ""
    :cText  := "Paisajes"
    :lGroup := .T.
```

```

END WITH
// Utilizando directamente :AddItem
:AddItem("L1","Lago 1",,"images\lago.jpeg")
:AddItem("L2","Lago 2",,"images\sample-1.jpg")
:AddItem(,"Entornos",.T.)
WITH OBJECT :AddItem()
  :cIndex := "E1"
  :cText  := "Oficina"
  :cImage := "images\office.jpg"
END WITH
:AddItem(,"Personas",.T.)
WITH OBJECT :AddItem()
  :cIndex := "P1"
  :cText  := "Yuna"
  :cImage := "images\yuna.jpg"
END WITH
:Create()
END WITH

```

Y para recuperar la selección ...

```

LOCAL cSeleccion := oCGI:GetCgiValue("selimagen")

```

Modo 4

Selección Múltiple

```

WITH OBJECT WComboBox():New(:WO)
  :cId      := "multiple"
  :aItems   := {"Enero","Febrero","Marzo","Abril","Mayo","Junio",;
"Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"}
  :cText    := "Meses"
  :cIcon    := "event_available"
  :cTitle   := "<b>Modo 4:</b> Selección Multiple de Mes"
  :cHelp    := "Meses en los que se realizarán las visitas periódicas"
  :lMultiple := .T.
  :Create()
END WITH

```

Y para recuperar la selección ...

```

LOCAL aSeleccion := oCGI:GetCgiArray("multiple")
// Como es una selección múltiple de esta manera nos retorna siempre un
array,
// aunque solo se seleccione un elemento.
LOCAL aSeleccion := oCGI:GetCgiValue("multiple")
// Si lo pedimos así, cuando se seleccionen varios elementos nos devolverá

```

```
un array
// si es una selección simple nos devuelve una cadena.
```

Propiedades

| Nombre | Init | Descripción | Versión |
|--------------|------|--|---------|
| cText | | Texto que aparece cuando no hay nada seleccionado | 0.1 |
| cTitle | | Título encima del ComboBox | 0.1 |
| clcon | | Icono que aparecerá a la izquierda del control | 0.1 |
| altems | | Elementos del ComboBox, se puede utilizar de 3 maneras | 0.1 |
| lMultiple | | Se permite la selección múltiple | 0.1 |
| cSelected | | cIndex seleccionado por defecto | 0.1 |
| cOnChange | | Función del CGI a llamar cuando cambie la selección, si empieza por # llama a función JS | 0.1 |
| aParams | | Parámetros extras a enviar cuando se llame a cOnChange | 0.1 |
| cAjaxBevel | | wBevel donde se incrusta el resultado de la llamada al Ajax | 0.1 |
| lAjaxWaitON | .T. | Incluye un indicador de actividad mientras se ejecuta la función Ajax | 0.1 |
| cAjaxText | | Mensaje a poner debajo del Reloj de Actividad | 0.1 |
| lReload | .F. | Cuando se envía por Ajax ha de activarse la Recarga para que materialize lo inicialice | 0.1 |
| lHtmlDefault | .F. | Fuerza a utilizar el Select por defecto de HTML y no el de Materialize, es útil para los navegadores IOS ya que el de Materialize no funciona aun correctamente. | 0.1 |

Métodos

| Nombre | Retorna | Descripción | Versión |
|--|------------|--|---------|
| AddItem(cIndex, cText, lGroup, clmage) | Nuevo Item | Añadir wItemComboBox a altems. | 0.1 |

Hereda de...

wControl

Propiedades

| Nombre | Init | Descripción | Versión |
|----------|--------|--|---------|
| cClassId | | Identificador del control para generar el Id | 0.1 |
| clId | Nil | Identificador único para cada Control | 0.1 |
| oParent | | Control contenedor | 0.1 |
| oHtml | | Objeto de la clase wHtml con el código para su renderizado | 0.1 |
| oStyle | | Objeto de la clase wStyle con las variantes de estilo | 0.1 |
| lEnabled | .t. | El control aparece Habilitado o Inhabilitado | 0.1 |
| lHide | .f. | El control aparece Oculto o Visible | 0.1 |
| aWidth | {12,,} | Ancho del control según el tamaño de la pantalla | 0.1 |

| Nombre | Init | Descripción | Versión |
|--------------------------|-------|---|---------|
| aOffset | {,,,} | Margen izquierdo según el tamaño de la pantalla | 0.1 |
| cClrText | black | Color de Texto | 0.1 |
| cClrPane | white | Color del Fondo | 0.1 |

10/11/2024 12:05

Métodos

| Nombre | Descripción | Versión |
|--------------------------|---|---------|
| Create | Este es el método que crea cada uno de los controles, esta desarrollado en cada Control | 0.1 |
| FullHtml | Nos devuelve una cadena con todo el código HTML y JS necesario para crear el control | 0.1 |
| LogDebug | Nos Añade un mensaje a la consola del navegador cuando estamos en modo Debugger | 0.1 |
| Assert | Nos Añade un comentario en el HTML cuando estamos en modo Debugger | 0.1 |
| AddHTML | Nos Añade código HTML dentro del control | 0.1 |
| AddParam | Añade un parámetro a la propiedad aParams , hay que pasarle un array {"nombre","valor"} | 0.2 |

10/11/2024 12:05

From:
<https://nefele.dev/wiki/> - **Nefele Project**

Permanent link:
<https://nefele.dev/wiki/controles/wcombobox?rev=1586428399>

Last update: **21/07/2020 10:50**

