# wEdit

Este control nos permite la introducción de datos en el AppWeb para que luego pueda ser enviado a nuestro CGI, puede ser Alfanumérico, Numérico, Contraseña, Email, Memo y Autocompletado.

Como Edit Memo el alto es dinámico para que quepa el texto.

Este control puede también funcionar como Date y Time, pero creemos que es mas interesante utilizar wDatePicker y wTimePicker, que en el fondo heredan de wEdit pero añaden funcionalidades más visuales.

Si es parte de un wForm al hacer el submit, se utiliza la propiedad cName como parámetro que recibirá el CGI, si no se define esta propiedad se le asignara el cId.

## Ejemplo de uso

```
WITH OBJECT WEdit():New(:WO)
  :cId           := "departamento"
  :cTitle        := "Departamento (cTitle)"
  :cHint         := "Indique departamento (cHint)"
  :cIcon         := "folder_shared"
  :SetAutocomplete({"Comercial","Autoventa","Preventa","Postventa",;
                    "Administración","Contabilidad","Recursos Humanos",;
                    "Dirección"})
  :SetRequired()
  :Create()
END WITH
WITH OBJECT WEdit():New(:WO)
  :cId           := "nombre"
  :cTitle        := "Nombre (cTitle)"
  :cIcon         := "person"
  :nLength       := 25
  :SetRequired()
  :Create()
END WITH
WITH OBJECT WEdit():New(:WO)
  :cId           := "password"
  :cTitle        := "Password"
  :cIcon         := "lock"
  :cHelp         := "Recuerde usar siempre contraseñas seguras (cHelp)"
  :nLength       := 10
  :SetPassword()
  :SetRequired()
  :Create()
END WITH
```

# Propiedades

| Nombre | Init | Descripción | Versión |
|---|---|---|---|
| cClassId | edt | | 0.1 |
| cIcon | | Icono junto al área de escritura, se puede elegir uno de la lista de Materialize | 0.1 |
| cName | | Nombre de la propiedad que recibe el CGI y que podrá leerse con oCGI:GetCgiValue | 0.1 |
| nStyle | xc_Text | Estilo de la Caja de Texto (ver en constantes) | 0.1 |
| lAutoFocus | .f. | Fuerza a que sea el control que reciba el foco al cargar la página | 0.1 |
| lRequired | .f. | Es un campo obligatorio | 0.1 |
| lSubmit | .f. | Si es **.t.** un intro en este campo causa que se pulse sobre el botón submit | 0.1 |
| cHint | | Mensaje que aparece en el control cuando esta vació y no tiene el foco<br>Si está vacío pero existe cTitle, mostrará este último en su lugar | 0.1 |
| cTitle | | Texto que aparecerá en la cabecera del control | |
| cHelp | | Texto de Ayuda debajo del control | 0.1 |
| lValidate | .f. | Fuerza la validación del tipo de dato (xc_Numeric y xc_Email) | 0.1 |
| lReadOnly | .f. | Activa el modo solo lectura | 0.1 |
| nLength | 0 | Longitud del campo 0 = indefinido si se indica una longitud no se dejarán introducir más que esos caracteres | 0.1 |
| nMinValue | 0 | Valor mínimo admitido (solo xc_Numeric) | 0.1 |
| nMaxValue | 9999999999999 | Valor máximo admitido (solo xc_Numeric) | 0.1 |
| aAutoComplete | {“”} | Lista de valores a buscar en caso de xc_Autocomplete (**1**) | 0.1 |

```
1.- Puede ser un array simple de elementos {"elem1","elem2",...},
    un array de array de elementos {{"elem1"},{"elem2"},...} que es lo que
nos devuelve QueryArray,
    o un array de array de Pares
{{"elem1","imagen1"},{"elem2","imagen2"},...}
    para incluir una imagen a la derecha de cada elemento
```

# Metodos

| Nombre | Descripción | Versión |
|---|---|---|
| SetNumeric | Define el control como Numérico | 0.1 |
| SetPassword | Define el control como Password | 0.1 |
| SetEmail | Define el control como Email | 0.1 |
| SetMemo | Define el control como Memo | 0.1 |
| SetText | Define el control como Texto | 0.1 |
| SetAutocomplete(aAutoComplete) | Habilita el modo AutoCompletar y asigna el array de elementos (**1**) | 0.1 |

| Nombre | Descripción | Versión |
|---|---|---|
| SetRequired | Asignación rapida de lRequired | 0.1 |
| SetSubmit | Asignación rapida de lSubmit | 0.1 |
| SetValidate | Asignación rapida de lValidate | 0.1 |

## Hereda de...

# wControl

## Propiedades

| Nombre | Init | Descripción | Versión |
|---|---|---|---|
| cClassId | | Identificador del control para generar el Id | 0.1 |
| cId | 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 |
| 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/wedit?rev=1578578355**

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