

## Menu

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Menu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 19, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Menu</b>	<b>1</b>
1.1	Menu V1.00 . . . . .	1
1.2	attachmenu . . . . .	1
1.3	createmenu . . . . .	2
1.4	initmenu . . . . .	2
1.5	menutitle . . . . .	2
1.6	menuitem . . . . .	3
1.7	menusubitem . . . . .	3
1.8	menubar . . . . .	3
1.9	menusubbar . . . . .	3
1.10	setmenucolour . . . . .	4
1.11	freemenu . . . . .	4
1.12	menucheckitem . . . . .	4
1.13	menuchecksubitem . . . . .	4
1.14	menutoggleitem . . . . .	5
1.15	menutogglesubitem . . . . .	5

# Chapter 1

## Menu

### 1.1 Menu V1.00

PureBasic Menu library V1.00

Le menu est un des éléments majeurs d'une interface graphique moderne. Avec le PureBasic, vous allez pouvoir les exploiter au maximum, en toute simplicité. Les menus créés utilisent automatiquement la propriété 'NewLook' du Workbench 2.0+

Commandes disponibles:

- AttachMenu
- CreateMenu
- FreeMenu
- InitMenu
- MenuBar
- MenuCheckItem
- MenuCheckSubItem
- MenuItem
- MenuTitle
- MenuSubBar
- MenuSubItem
- MenuToggleItem
- MenuToggleSubItem
- SetMenuColour

Exemple:

Menu demo

### 1.2 attachmenu

Syntaxe

`AttachMenu (#Menu, WindowID)`

Résumé

Attache le menu spécifié à la fenêtre donnée.

Si vous voulez enlever le menu de la fenêtre, vous devez utiliser la commande `DetachMenu()` de la bibliothèque 'Window'.

**#Menu:** Identifiant numérique du menu à attacher à la fenêtre

**WindowID:** Identifiant de la fenêtre sur laquelle le menu doit être attaché. Il est conseillé d'utiliser la commande `WindowID()` pour obtenir cet identifiant.

### 1.3 createmenu

Syntaxe

`CreateMenu(#Menu, ScreenID)`

Résumé

Crée réellement le menu qui est en train d'être construit.

**#Menu:** Identifiant numérique du menu qui va être créé.

**ScreenID:** Identifiant de l'écran sur lequel le menu va être affiché. Il est conseillé d'utiliser la commande `ScreenID()` pour obtenir cet identifiant.

### 1.4 initmenu

Syntaxe

`InitMenu(#MaxMenus, #MaxMenuItems)`

Résumé

Initialise l'environnement nécessaire à la gestion des menus. Vous devez appeler cette fonction avant d'appeler une autre fonction de cette bibliothèque.

**#MaxMenus:** Nombre maximal de menus à gérer

**#MaxMenuItems:** Nombre maximal d'éléments par colonne que peut avoir un menu.

### 1.5 menutitle

Syntaxe

`MenuTitle(Title$)`

Résumé

Crée un nouveau titre pour le menu en cours de construction.

---

Title\$: Titre qui figurera dans le menu.

## 1.6 menuitem

Syntaxe

```
MenuItem(MenuID, Text$, ShortCut)
```

Résumé

Crée un élément pour le menu en cours de construction.

MenuID: Identifiant numérique de l'élément qui va être créé.

Text\$: Nom de l'élément qui sera affiché dans le menu

ShortCut: Lettre qui déterminera le raccourci pour cet élément.

## 1.7 menusubitem

Syntaxe

```
MenuSubItem(MenuID, Text$, ShortCut)
```

Résumé

Crée un sous-élément pour le menu en cours de construction.

MenuID: Identifiant numérique de l'élément qui va être créé.

Text\$: Nom de l'élément qui sera affiché dans le menu

ShortCut: Lettre qui déterminera le raccourci pour cet élément.

## 1.8 menubar

Syntaxe

```
MenuBar()
```

Résumé

Crée une barre de séparation dans le menu.

## 1.9 menusubbar

Syntaxe

```
MenuSubBar()
```

Résumé

Crée une barre de séparation dans le sous-menu.

---

## 1.10 setmenucolour

### Syntaxe

```
SetMenuColourCcolour)
```

### Résumé

Change la couleur du menu par la couleur spécifiée. Cette commande doit être appelée avant `CreateMenu()` .

Colour: Couleur à utiliser pour le menu

## 1.11 freemenu

### Syntaxe

```
FreeMenu(#Menu)
```

### Résumé

Detruit le menu spécifié. Vous ne pourrez plus utiliser ce menu ultérieurement. Vérifiez que le menu que vous détruisez n'est plus attaché à aucune fenêtre.

## 1.12 menucheckitem

### Syntaxe

```
MenuCheckItem(#Item, Label$, ShortCut$, State)
```

### Résumé

Ajoute un élément de type 'Check' à la suite du menu en cours de construction.

MenuID: Identifiant numérique de l'élément qui va être crée.

Text\$: Nom de l'élément qui sera affiché dans le menu

ShortCut\$: Lettre qui determinera le raccourci pour cet élément.

State: Etat initial du 'Check Menu' (1=activé, 0=Désactivé)

## 1.13 menuchecksubitem

### Syntaxe

```
MenuCheckSubItem(#Item, Label$, ShortCut$, State)
```

### Résumé

Ajoute un élément de type 'Check' à la suite du sous-menu en cours de construction.

---

MenuID: Identifiant numérique de l'élément qui va être créé.

Text\$: Nom de l'élément qui sera affiché dans le menu

Shortcut\$: Lettre qui déterminera le raccourci pour cet élément.

State: Etat initial du 'Check Menu' (1=activé, 0=Désactivé)

## 1.14 menutoggleitem

### Syntaxe

MenuToggleItem(#Item, Label\$, Shortcut\$, State)

### Résumé

Ajoute un élément de type 'Toggle' à la suite du menu en cours de construction.

MenuID: Identifiant numérique de l'élément qui va être créé.

Text\$: Nom de l'élément qui sera affiché dans le menu

Shortcut\$: Lettre qui déterminera le raccourci pour cet élément.

State: Etat initial du 'Toggle Menu' (1=activé, 0=Désactivé)

## 1.15 menutogglesubitem

### Syntaxe

MenuToggleSubItem(#Item, Label\$, Shortcut\$, State)

### Résumé

Ajoute un élément de type 'Toggle' à la suite du sous-menu en cours de construction.

MenuID: Identifiant numérique de l'élément qui va être créé.

Text\$: Nom de l'élément qui sera affiché dans le menu

Shortcut\$: Lettre qui déterminera le raccourci pour cet élément.

State: Etat initial du 'Toggle Menu' (1=activé, 0=Désactivé)

---