



مدخل الى لغة

بيوربيزك

أكتب بقوة السّي بلس+ و سرعة الأسمبلي



عبدالهادي جعفر

@nodekits

مدخل الى لغة بيوربازك

أكتب بقوة السّي بلس + و سرعة الأسمبلي

PureBasic – The Perfect Cross-Platform & Native Development Language

Abdelhadi
@nodekits

عبدالهادي جعفر .. مُسلم جزائري، طالب سنة أولى جامعي

تخصّص علوم حاسب، مهتم بالتصميم و برمجة التطبيقات [سطح المكتب / الويب].

أهديه الى .. شركة صخر (عسى أن تصل التحية) / مُجمّع حسوب و مواقع الكُتب العربية.

مقدمة الى Purebasic

لغة **بِيُور بِيُزِك Purebasic** أو اختصاراً بـ **PB** هي لغة برمجية و بيئة تطويرية متكاملة، تقوم على أسس لغة **Basic**، قامت بتطويرها و تصميمها شركة **fantaisie software** لتكون عابرة للمنصات و تدعم نُظْم التشغيل: **الويندوز و لينوكس** بمعيارية **32/64-bit** اضافة الى نظام **الماكنتوش**؛ كما أنها أتاحت نسخة مفتوحة المصدر قديمة نسبياً (النسخة 4) لنظام **Amiga** أيضاً!

عادةً ما يُشار الى رمز اللغة بحرف **Z**، أو قديماً بالبرغي.
صيغ ملفات اللغة: **.pb, .pbi, .pbf, .pbp, .pbv**

تم تصميم اللغة في أواخر التسعينات على نظام **AmigaOS** ، و أول اصدار عام للغة **Purebasic** شَمَل نُسخة ويندوز كان في 17 ديسمبر 2000. و يتم تحديثها باستمرار منذ ذلك الحين. و تعتمد اللغة عند اقتنائك على نسخة منها لترخيص (**مدى الحياة**) أو ما يعرف بـ "**LifeTime License Model**" و كما ورد من موقعهم أن هناك أشخاصا مازال لديهم حق الوصول للتحديث منذ عام 1998 (:

استطاعت لغة بيوربيزك لَمَّ و استنباط جميع خصائص اللغات الأخرى و توظيفها، **كقوة لغة السي**، و **الجافا**، و **لازاروس** .. و أيضا **#C** ؛ و **صِغَر** و **سرعة** و **حَجْم** ملفات التنفيذ كأنك تكتب بلغة التجميع (**Assembly**)؛ و سهولتها (**Syntax**) **كالبايثون و الروبي**؛ كما يمكنك بناء ألعاب ثلاثية الأبعاد باستخدام **DirectX** و غيرها من مشغلات الألعاب؛ زيادة الى بناء برامج تعمل على نواة النظام (**Kernel Mode Driver**).

لتقول لك اجمالاً: أكتب بخواص جميع اللغات .. **بلغة واحدة** .. و .. **سطر واحد** ..

*يمكنك تحميل نسخة تجريبية من الموقع الرسمي لديهم.

لما كل هذا التميز! ما الجديد فيها ؟

لغة Purebasic ليست لغة برمجة عادية، فقد سبق أن برمجت تطبيقات عديدة، مستعملاً لغات مختلفة كالسي و لازاروس و أوتويت Autoit لذلك أدرك تماماً قيمة هته اللغة و مستقبلها الشاسع.

بطبع الحال، بيوربيزك ليست أحسن لغة؛ ف C# أوسعُ منها في بناء تطبيقات الويندوز، و البايثون و الجافا في تعدد المنصات؛ لكن أن تجد لغة طبيعية + لا تحتاج الى طرف ثالث (dll's) + متعددة المنصات + سريعة العمل و خفيفة حجم + برمجة Kernel Driver فذاك شئ يستحق الثناء.

و لتبسيط أكثر: تولد تطبيقات طبيعية (Native Development) مستقلة، لا تحتاج الى طرف ثالث أو مكتبات خارجية مثل لغات .NET التي تُلزمك بتحميل مكتبات اضافية .. أو الجافا بمفسرها VM و كما أن شفرة برمجية واحدة تدعم جميع المنصات الأخرى (Cross-Platform). و يمكنك استدراج شفرات لغة التجميع (الأسمبلي) بكل يسر و سهولة inline asm .. كما أنك قادر على تطوير و برمجة مُتحكمات لوحة الأردوينو Arduino.

جميل جداً، لكن أين السهولة في العمل؟

لنأخذ مثالا بسيطاً، تميزت لغة ++C/C ببعويتها عند التعلم، و خصوصاً عند استخدام الواجهات الرسومية (GUI) فكانت تُلزمك باستعمال شيينين :

1 - اما استعمال أوامر النظام و ملحقاته OS API ك WinApi32 و GDK/GTK على لينوكس.

2 - أو استعمال برامج و مكتبات خارجية مكلفة و صعبة ك QT .. الخ.

أما PB's فسهولة أخرى سنعرفه لاحقاً.

الترجم Compiler لـ PB's

مُترجم لغة بيوربيزك صُمم ليُدعم جميع أصناف المنصات بمختلف المعالجات: x86, x86-64, PowerPC و 680x0

تتم أول ترجمة للملف بتجميعه و تحويله من صيغة PB الى شفرات أسمبلي (نوع Fasm) ثم يقوم مترجم الأسمبلي بتحويلها بشفرات مُضمنة الى ملف تنفيذي EXEs بحجم صغير جداً (light-weight) يصل الى ٢ كيلوبايت.

يستدعي على الويندوز ملف msvcrt.dll في حالة استخدامك للملحقاته API.

مكتبة PB's القياسية

و لنغوص أكثر ضمن اللغة، تستند بيوربيزك على مكتبات قياسية، تحوي وظائف كثيرة للاستخدام كمثال:

- صيغ BMP، JPEG، JPEG2000، PNG، TGA، TIFF.
- محرر/مُتحكم الشفرات البرمجية Scintilla.
- قواعد البيانات بجميع أنواعها SQLite، PostgreSQL، ODBC.
- محرك ألعاب مُصمَّن داخل اللغة Ogre 3D engine يمكنك انشاء رسومات ثنائية و ثلاثية الأبعاد و اعتماد بشكل غير رسمي لمحرك Irrlicht Engine.
- مكتبة DirectX المرفقة بالبيئة .. الخ.

بناء الجمل Syntax

بناء الجمل و اجراءاتها في بيوربيزك سهل للمُبتدئين و سريع للمتقدمين، و غير مكلف. تتشابه مع بعض صيغ فيجوال بيسك؛ كما تم تطويرها في النسخة 5.20 لتدعم بذلك البرمجة الكائنية OOP

لنلق نظرة حول شفرات اللغة:

#التعليقات - Comments

```
Code() ; Comment as in the asm
```

#المتغيرات - Variables

انشاء مُتغير رقمي صحيح و تعريفه، حيث أن i تعني كلمة Integer.

```
x.i
x.i = 5
```

لاستعمال المعاملات الحسابية بين المتغيرات ..

```
x.i
x = 5

y.i
y = x * 2 ; 10
```

لتعريف متغيرات نصية يمكننا ذلك بطريقتين: نستعمل الحرف (s) أو (\$)

```
str1.s = "Abdelhadi"           ; 1 way
str2$  = "nodekits"           ; 2 way

str3$ = str1 + " " + str2$ ; Abdelhadi nodekits
```

يمكن تحديد قيمة المتغير و تثبيته:

```
str1.s{5}
```

#الثوابت - Constants

```
#x = 1
#y = "2"
#z = 3.14
```

#العداد - Enumerations

```
Enumeration
#x ; 0
#y ; 1
#z ; 2
EndEnumeration
```

#نافذة طرفية - Console Interface

```
OpenConsole()
Print("Hello world!")
Input() ; wait for key or input a value
```

#نافذة المنقح - Debug-time console interface

تفيد في تنقيح بعض المشاكل و الأخطاء التي ضمن مشاريعك بدّل ترجمتها كلها، و تعطيك الوقت الحقيقي للتنفيذ ..

```
Debug "It will be printed in the debug output window"
```

#الاجراءات - Procedures

```
Procedure MsgBox(text$)
  MessageRequester("", text$)
EndProcedure

MsgBox("Hello World")
```

و لارجاع قيمة متغير داخل الإجراء:

```
Procedure$ InputBox(text$)
  ProcedureReturn InputRequester("", text$, "")
EndProcedure

name$ = InputBox("Print your name")
MessageRequester("", name$)
```

#التوجيه بلغة التجميع inline asm

من جماليات اللغة أتاحت استعمال شفرات لغة التجميع (أوامر الأسمبلر x86) و تضيفها ضمن برمجياتك عبر المنصات مباشرة، **بالمغيرات أو المؤشرات**؛ تُفيد الأسمبلي بشكل كبير ان كان لديك مشروع يستهلك أجزاء كبيرة من الذاكرة، بإمكانك استعمال شفرات أسمبلي لتسريع المهمة و الأداء لأنها تترجم مباشرة الى لغة الآلة.

على الويندوز و لينوكس بيوربيزك تستعمل الأسمبلر **Fasm** (<http://flatassembler.net>) و في نظام الماكنتوش تستخدم الأسمبلر **Yasm** (<http://yasm.tortall.net>) كما يمكنك الاستعانة بالمستندات و الوثائق التي يقدمها كلتا الموقعين.

من أجل تفعيل **التوجيه** علينا وضع الوسمين في البداية و النهاية ليتعرف عليهما المترجم.

- EnableASM : بداية التوجيه و تبليغ المترجم.

- DisableASM : نهاية التوجيه ..

لاستخدام صحيح للأوامر علينا تعريف **المتغيرات** أو **المؤشرات** قبل استعمالهما ضمن التعليمات، و ان استعملت ال (module) يجب كتابة اسمه بحروف صغير هكذا [**module_name.**].

مثال:

```
MOV ebx, ll_mylabel
...
MyLabel:
```

.. ملاحظات ..

- الأخطاء التي تظهر في التوجيه تابعة للأسمبلر و ليس للبيوربيزك، قم بمراجعة الخطأ و استخدم التوثيق المرفقة.

1- لا يمكن استعمال الكلمات المفتاحية للأسمبلر **keywords** خارج التوجيه، ضمن أسماء Lable.

2- معالجات x86 تُستعمل الكلمات المحجوزة ك: **eax**، **ecx**، **edx**؛ الى آخره من المفاتيح.

3- معالجات x64 تستعمل الكلمات **rax**، **rcx**، **rdx**، **r8**، **r9**، **xmm0**، **xmm1**، **xmm2**، **xmm3** .. الى آخره من المفاتيح.

4- على الويندوز فقط: ملف المساعدة (ASM help-file) يمكن تحميله على [الرابط التالي](#) و قم بنقل ملف 'ASM.HLP' الى المجلد في البيوربيزك 'Help/' ليتمكنك الاستعانة بالنقر على الكلمة المحجوز و الضغط على الزر **F1** لاطهار تفاصيله.

5- للعودة مباشرة الى المسجل **eax** أو (**rax** في معمارية x64) استعمل فقط **ProcedureReturn** بدون أي عبارة أخرى.

أمثلة:

```
Procedure Test(*Pointer, Variable)
! MOV dword [p.p_Pointer], 20
! MOV dword [p.v_Variable], 30
Debug *Pointer
Debug Variable
EndProcedure
```

```
Procedure.l MyTest()
MOV eax, 45
ProcedureReturn ; The returned value will be 45
EndProcedure
```

#البيانات في الملفات التنفيذية EXE

عادة عند بناء تطبيقات ذو واجهة رسومية يمكن للقراصنة استخدام برامج تقوم بفك التجميع، و تعرض بيانات الأخير ك [ResHacker](#) ، لكن على بيوربيزك لا يمكن ذلك. كمثال:

— البيانات العددية:

```
Restore NumData
Read.l A
Read.l B

MessageRequester("", Str(A))
MessageRequester("", Str(B))

DataSection
  NumData:
  Data.l 1, 2
EndDataSection
```

— البيانات النصية:

```
Restore NumData
Read$ A$
Read$ B$

MessageRequester("", A$ + B$)

DataSection
  NumData:
  Data$ "Hello ", "world"
EndDataSection
```

— اظهر محتوى ملف نصي:

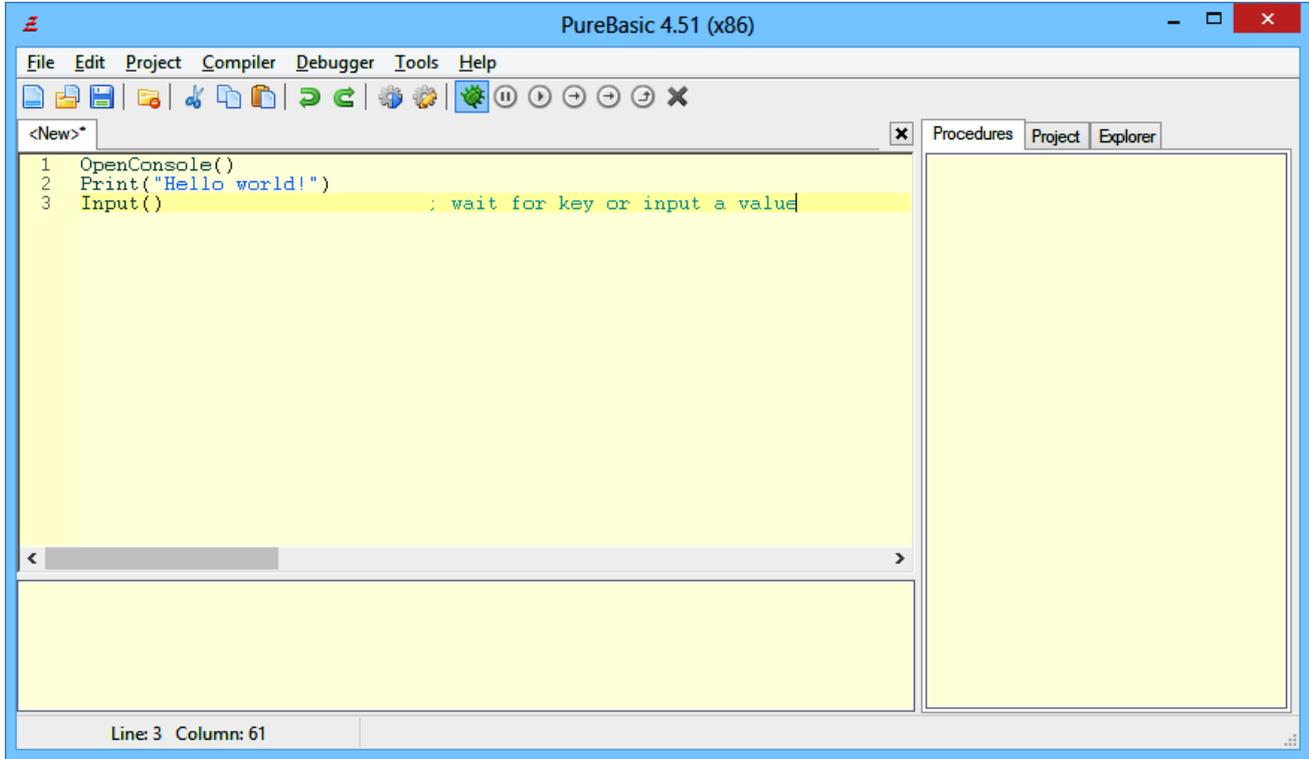
```
Restore File1
Read.s File1Data$

MessageRequester("", File1Data$)

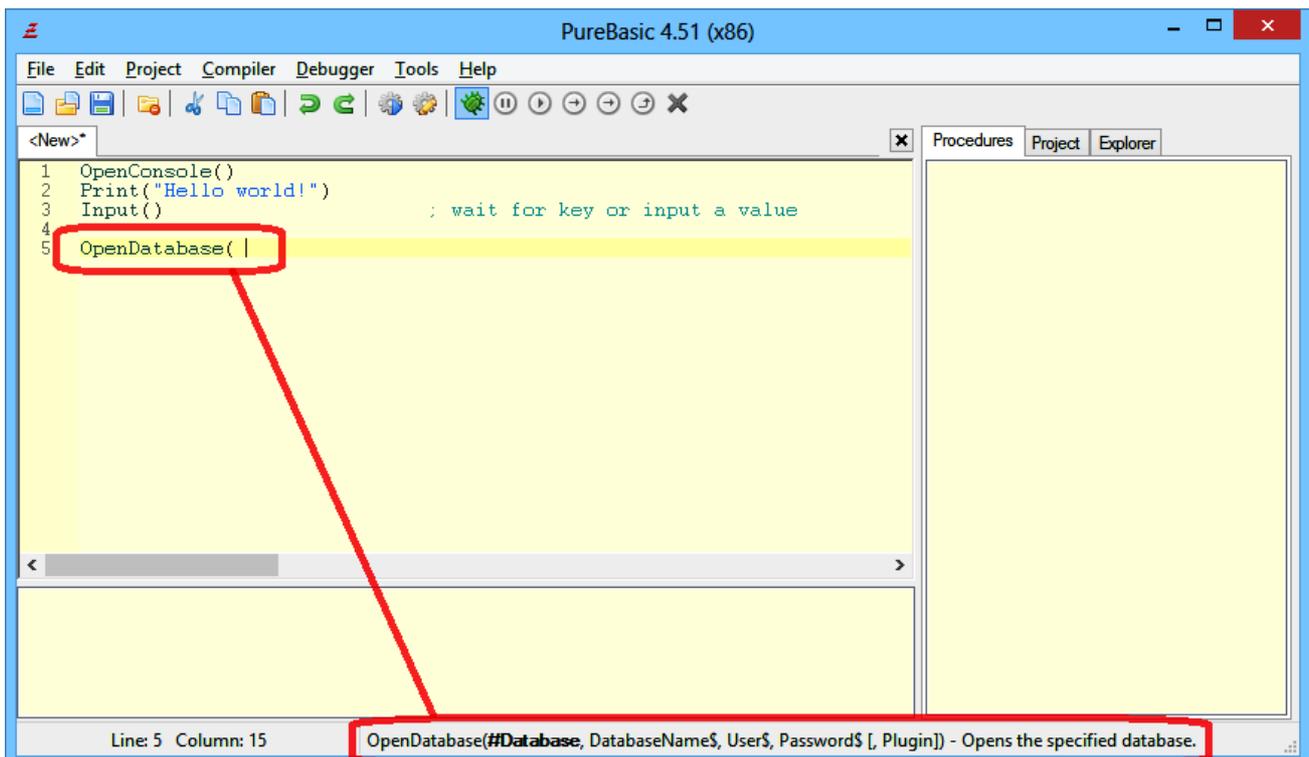
DataSection
  File1:
  IncludeBinary "file.txt"
EndDataSection
```

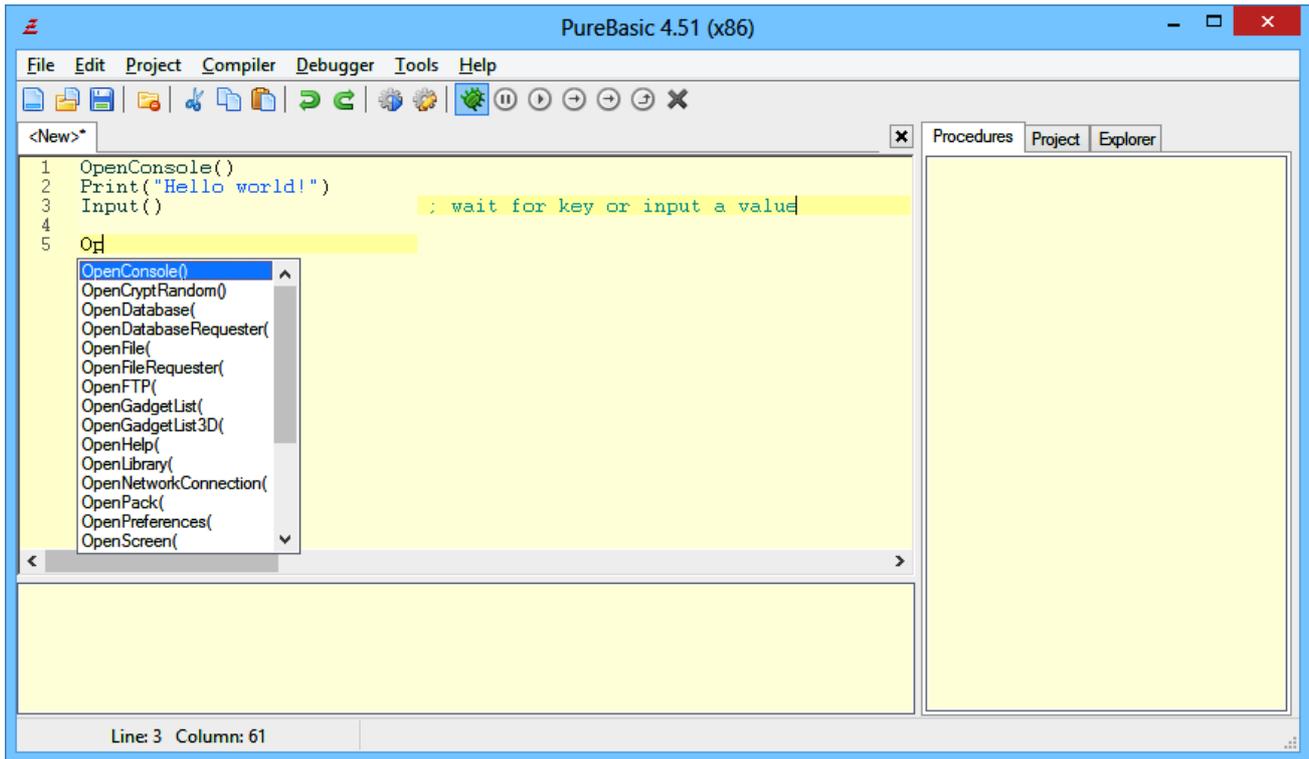
نظرة حول البيئة التطويرية IDE

يأتي مع اللغة بيئة تطويرية متكاملة لانشاء مشاريع برمجية كبيرة، تحوي هذه المحررات على خصائص كبيرة مثل تنقيح الشيفرات البرمجية، و ادارة المشاريع و تصنيفها، بيئة بيوربيزك بسيطة جداً و غير معقدة تميزت بسهولة الكبيرة في ادارة خصائص المحرر، و خيارات الاكمال التلقائي، تحرير الألوان، تخصيص المترجم، التنقيح التلقائي، الى جانب توثيق اللغة بالضغط على زر **F1** .. لنلق نظرة :

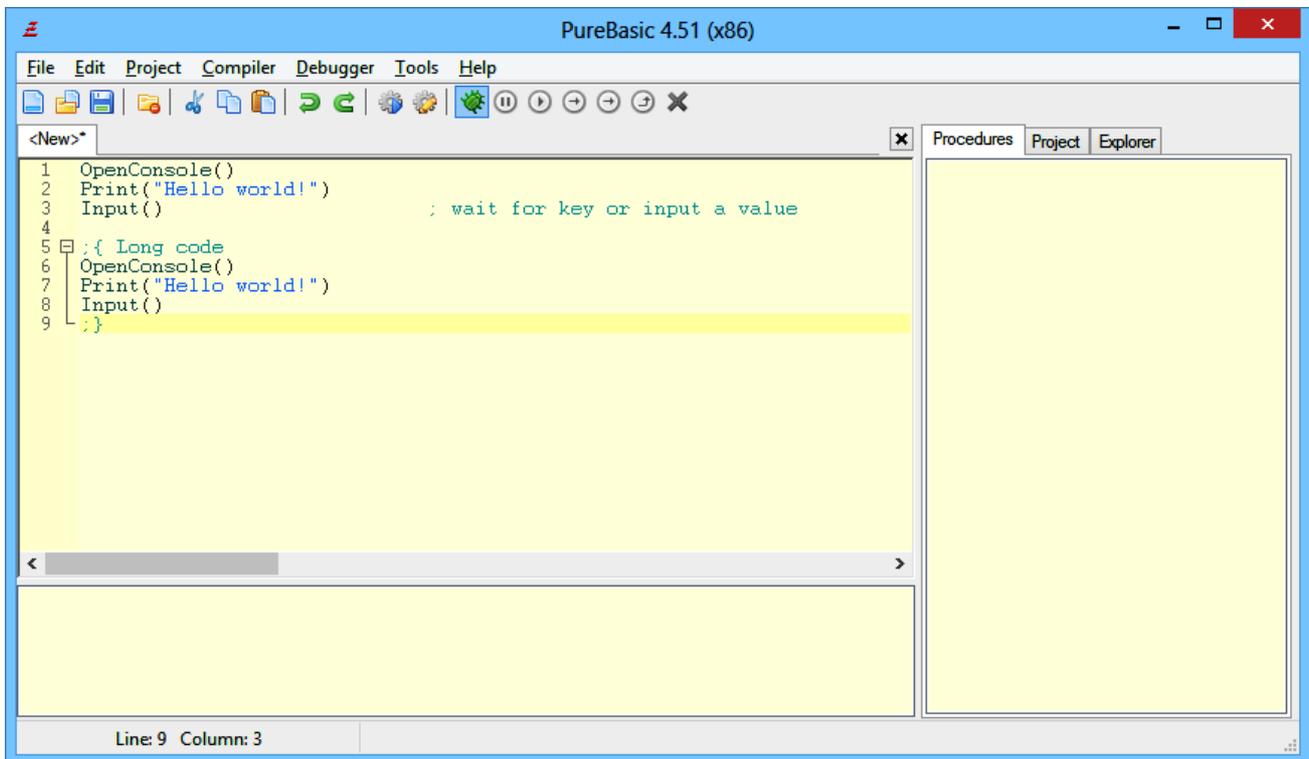


(الصور من النسخة 4 القديمة)

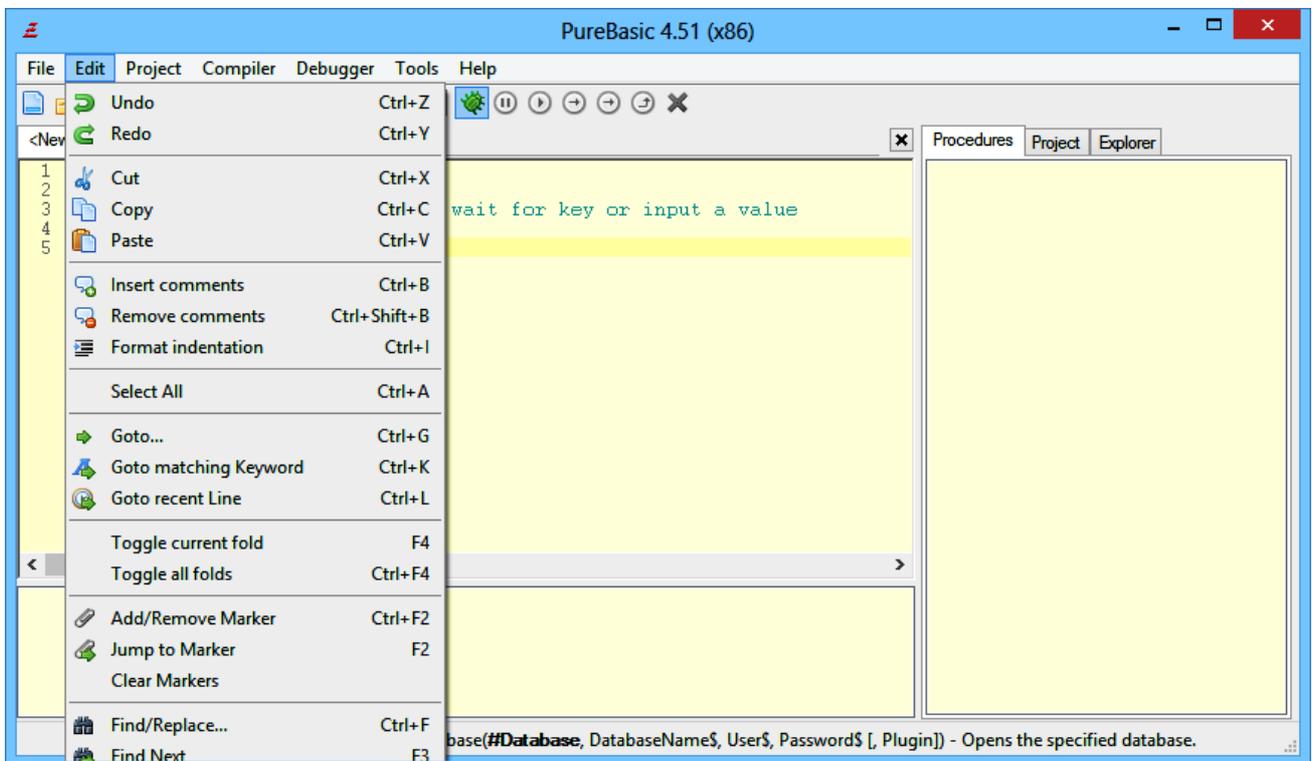
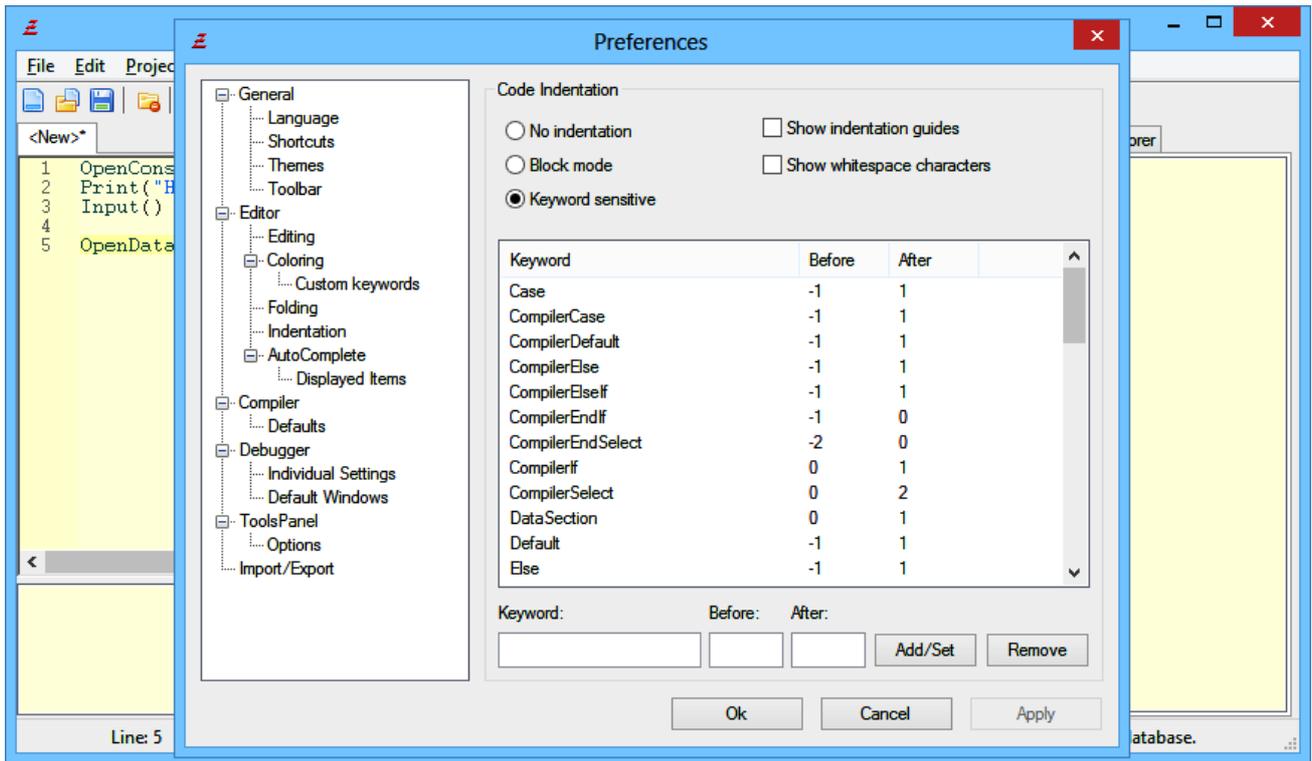




لاكمال العبارة اضغط زر **TAB** .. في محررات أخرى تستعمل زر **مسافة** ؛ يمكنك تخصيصها في الاعدادات.



خاصية **طي الشيفرات** و **تقليصها** .. لتقليل حجم الأسطر.



انشاء الواجهات الرسومية GUI

في لغة بيورك بيزك يمكنك انشاء واجهات رسومية بخمسة طرق:

- 1 - باستخدام المكتبة الأصلية ضمن PB فقط، و متعددة المنصات .. (الطريقة الأسهل).
 - 2 - باستخدام OS API كما ذكرنا سابقاً، هذه الطريقة تمكنك من التحكم بالنظام بشكل كامل، مع صغر حجم الملف.
 - 3 - الجمع بين 1 و 2 ان كنت تحتاج ذلك .. (الوسيلة المفضلة في معظم الحالات).
 - 4 - استخدام محرك 3D للتصميم و الرسم، و تطوير ألعاب ثلاثية الأبعاد تشبه الى حد ما WPF في الدوت نت .NET .. (الوسيلة المنصوح بها لتطوير ألعاب 3D).
 - 5 - استعمال الطُرق 1 و 2 و 4 .. (وسيلة جميلة لكن تعتمد على حالة المكتبات).
- #الطريقة 1: استخدام المكتبة الأصلية ..
واجهة رسومية بسيطة .. نافذة واحدة و نافذة متعددة ..

```
MessageRequester("Message Box", "Hello World") ; 4.5 KB
```

```
Enumeration
  #btn1
EndEnumeration

OpenWindow(#PB_Any, 10, 50, 400, 200, "Abdelhadi nodekits",
#PB_Window_SystemMenu)

ButtonGadget(#btn1, 10, 10, 100, 25, "Button 1", 0)

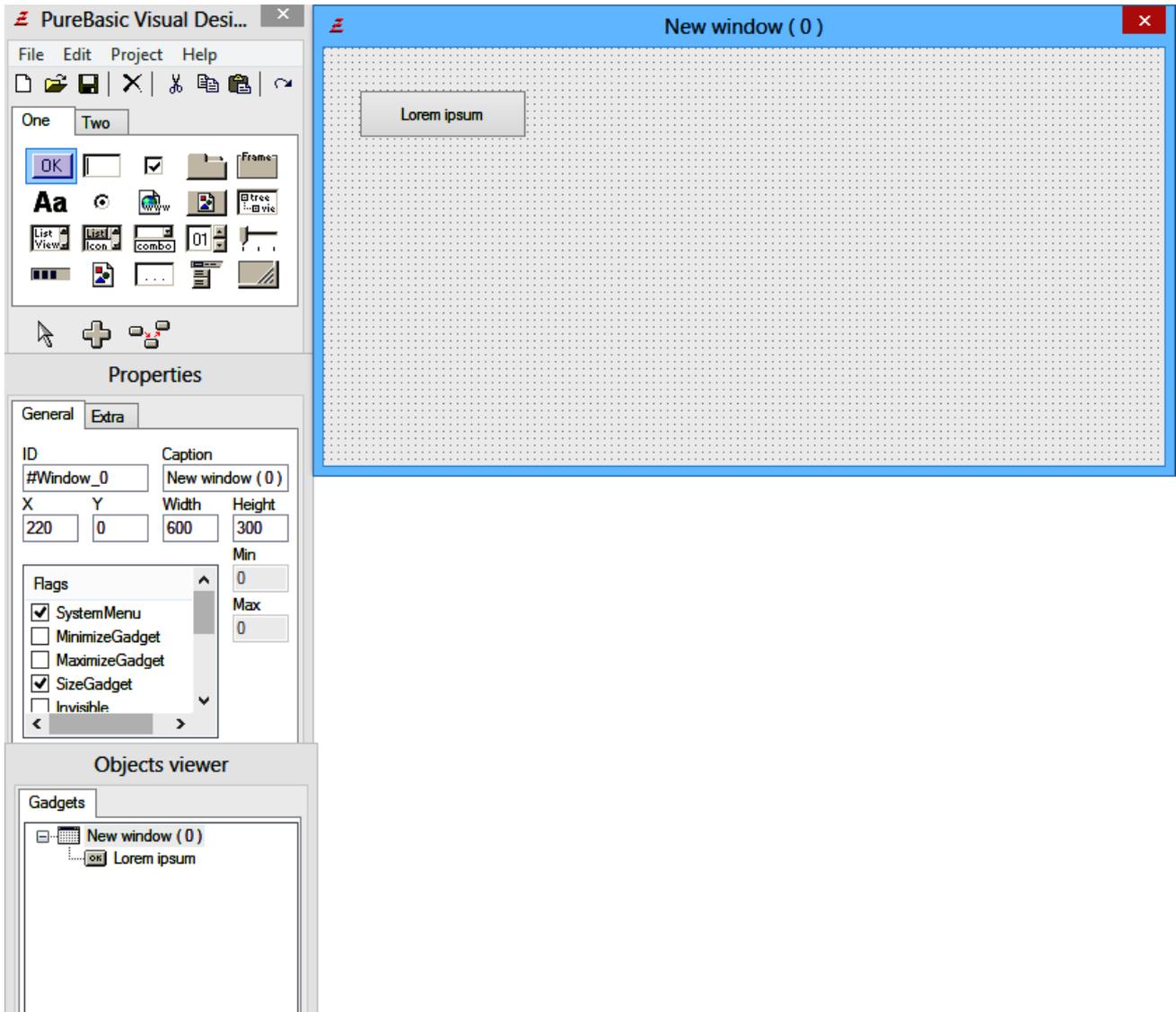
Repeat
  Event = WaitWindowEvent()

  If Event = #PB_Event_Gadget
    GadgetID = EventGadget()

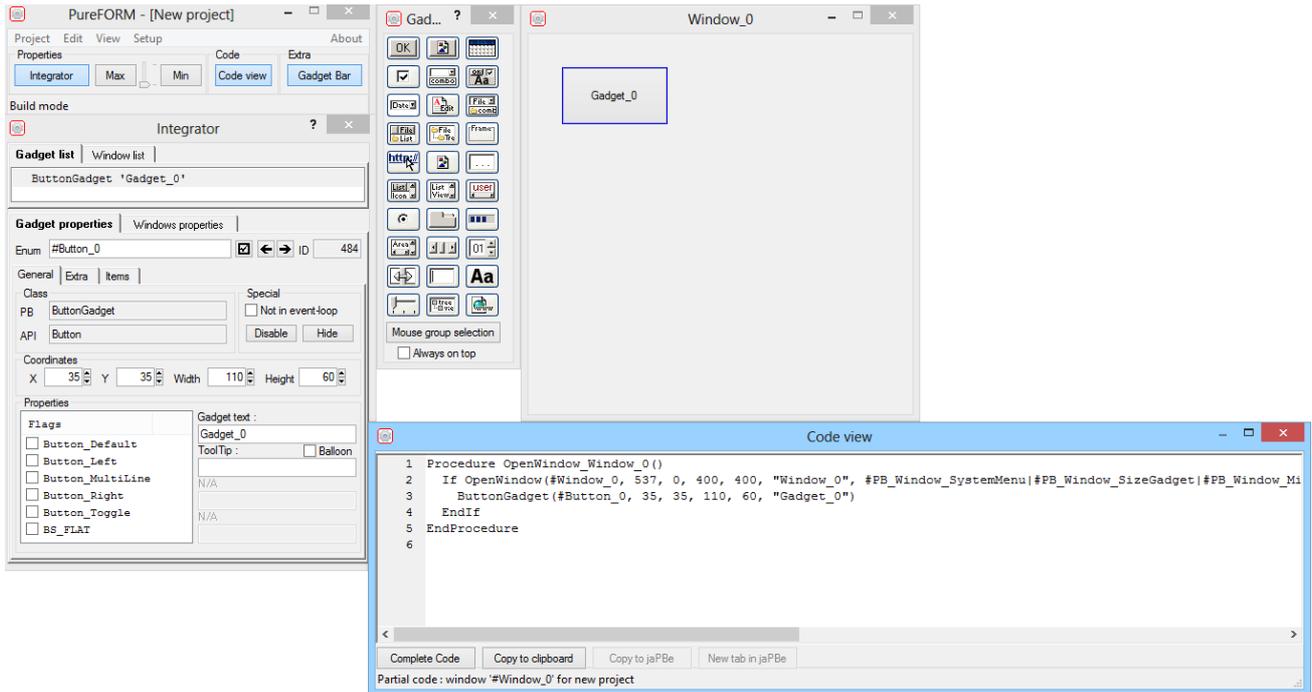
    If GadgetID = #btn1
      MessageRequester("", "Button 1 clicked!")
    EndIf
  EndIf

Until Event = #PB_Event_CloseWindow
```

هذه أسهل طريقة لكتابة واجهة رسومية بشكل يدوي على بمكتبة بيوربيزك، يمكنك الاستعانة بمصمم واجهات سريع جداً وفعال (**Visual Deigner**)، لتفعيله تجده في القائمة العلوية



أو يمكنك أيضا استعمال محرر آخر يدعى بـ **PureFORM editor** .. بعد تصميمك للبرنامج يعطيك خاصية نسخ كود التصميم لتستخدمه على البيئة التطويرية ..



.. في الأخير كل السُّبُل مُتاحة لتُنشئ برنامجك بكل ارتياحية و سهولة ..

#الطريقة 2: استخدام أدوات النظام..

باستعمال OS API تدعى على الويندوز WinAPI (في النسخة 10 تَغْيِر اسمها).

نافذة بسيطة بسطر واحد .. حجم الملف التنفيذي لا يصدّق "2.0 KB"

```
MessageBox_(0, "Hello World", "Message Box", 0) ; 2.0 KB
```

نوافذ متعددة ..

```
; If you want to use in GUI not only latin characters, compile it in Unicode!
; Warning: exe size - it's very, very small!
```

```
Declare.l WndProc(hWnd, Msg, wParam, lParam) ; declare Window events callback
```

```
; Global vars
WindowClass.s = "WndClass1"
```

```
; Initialize Window Class
wc.WNDCLASSEX
wc.cbSize = SizeOf(WNDCLASSEX)
```

```

wc\hbrBackground = #COLOR_WINDOW
wc\hCursor = LoadCursor_(0, #IDC_ARROW)
wc\lpfnWndProc = @WndProc()
wc\lpszClassName = @WindowClass

; register Window Class
If RegisterClassEx_(@wc) = 0
    MessageBox_(hWnd, "Can't register main window class.", "", #MB_ICONERROR)
End
EndIf

; create window
hWnd = CreateWindowEx_(0, WindowClass, "Abdelhadi Nodekits", #WS_SYSMENU,
10, 50, 400, 200, 0, 0, 0, 0)
If hWnd = 0
    MessageBox_(hWnd, "Can't create main window.", "", #MB_ICONERROR)
End
EndIf

; create button and set it's font
hButton1 = CreateWindowEx_(0, "Button", "Button 1", #WS_CHILD | #WS_VISIBLE,
10, 10, 100, 25, hWnd, 0, 0, 0)
SendMessage_(hButton1, #WM_SETFONT, GetStockObject_(#DEFAULT_GUI_FONT), 1)

; show window
ShowWindow_(hWnd, #SW_SHOWDEFAULT)
UpdateWindow_(hWndMain)

; messages handling loop
While GetMessage_(msg.MSG, #Null, 0, 0 )
    TranslateMessage_(msg)
    DispatchMessage_(msg)
Wend

; window events callback
Procedure.i WndProc(hWnd, Msg, wParam, lParam)
    Shared hButton1

    Select Msg
        Case #WM_COMMAND
            If hButton1 = lParam
                MessageBox_(hWnd, "Button 1 clicked!", "", #MB_OK)
            EndIf
        Case #WM_CLOSE
            DestroyWindow_(hWnd)
        Case #WM_DESTROY
            PostQuitMessage_(0) : Result = 0
        Default
            Result = DefWindowProc_(hWnd, Msg, wParam, lParam)
    EndSelect

    ProcedureReturn Result

```

البرمجة الكائنية OOP

مطور بيوربيزك سابقاً fred أكد أنه لن يدعم أبداً البرمجة الكائنية، لكن بعد بلوغ عدد كبير من مستخدمي اللغة أُجبرت على دعمها للبرمجة الكائنية مما لاقت استخداماً كبيراً بين المطورين. [راجع الرابط](#).

استخدام مكتبات الطرف الثالث - Third-party

تحتوي مكتبة بيوربيزك على مكتبات كبيرة مفتوحة المصدر تجدها على الرابط : <http://pboisl.purearea.net> كما يمكنك تزيين واجهاتك الرسومية عبر مكونات ProGUI يعطيك أكثر 100 ملف DLL مع مستندات و توثيق لتخصيصها بشكل أكبر؛ كما يتوفر الويب على ستايلات skins بأشكال مختلفة.

الملاحق و التوثيقات

جدول بسيط معرّب يوضح أنواع المتغيرات ..

النوع	الملاحقة	استخدام الذاكرة	مجال الأعداد
بايت	b.	1 بايت (8 بت)	127 + ... 128-
أسكي	a.	1 بايت (8 بت)	255 + ... 0
حرف	C.	1 بايت (8 بت) (أسكي)	255 + ... 0
كلمة	w.	2 بايت (16 بت)	32767 + ... 32768-
يونيكود	u.	2 بايت (16 بت)	65535 + ... 0
حرف	C.	2 بايت (16 بت) (يونيكود)	65535 + ... 0
طويل	L.	4 بايت (32 بت)	2147483647 + ... 2147483648-
عدد صحيح	i.	4 بايت (32 بت) إلى x86	2147483647 + ... 2147483648-
Float	F.	4 بايت (32 بت)	اعتماداً على نسبة من رقم عشري.
عدد صحيح	i.	8 بايت (64 بت) إلى x64	9223372036854775807 + ... 9223372036854775808-
رباعية	q.	8 بايت (64 بت)	9223372036854775807 + ... 9223372036854775808-
Double	d.	8 بايت (64 بت)	اعتماداً على نسبة من رقم عشري.
نص	S.	طول سلسلة + 1 بايت	لا حدود.
سلسلة ثابتة	S. {طول}	طول سلسلة	لا حدود.

المنتديات ..

- (الإنجليزية) <http://www.purebasic.fr/english>
- (الألمانية) <http://forums.purebasic.com/german>
- (الفرنسية) <http://www.purebasic.fr/french>
- (الروسية) <http://purebasic.info/phpBB3ex/index.php>
- (الروسية) <http://www.cyberforum.ru/pure-basic>
- (الروسية) <http://purebasic.mybb.ru>
- (الروسية) <http://purebasic.ucoz.ru/forum>
- (الصينية) <http://www.purebasic.cn/forum.php>

وثائق و دروس + شفرات برمجية جاهزة ..

- <http://www.purebasic.com/documentation>
 - <http://purearea.net/pb/CodeArchiv/CodeArchiv.html>
 - (الروسية) <http://pure-basic.narod.ru>
 - (الروسية) <http://purebasic.info/Chapters/index.html>
 - (الروسية) <http://purebasic.ucoz.ru>
 - (الروسية) <http://mirashic.narod.ru>
 - (الروسية) <http://purebasic.ucoz.com>
- (رُبما بعض المواقع توقفت، استعن بموقع الأرشيف archive.org)

المراجع

- الموقع الرسمي للغة بيوربازيك purebasic.com
- The Perfect Cross-Platform .. etc [CodeProject](http://codeproject.com)
- موقع ويكيبيديا .. [Purebasic](http://purebasic.com)

أتمنى في الأخير أن يعجبكم هذا المدخل المختصر للغة، ربما اصراري على تعلمها و عدم وجود أي تعريف بها على الويب العربي ألزمني كتابة هذا الكتيب البسيط .. أرجوا من الله أن يكتب أجره في ميزان حسناتنا.
دعواتكم لنا بظهر الغيب .. أخوكم عبدالهادي.

Abdelhadi

@nodekits

الاميل: nodekits@gmail.com