

بناء واجهات المستخدم الرسومية في لغة بيرل Perl gui

مقدمة :

كمبرمج php طالما تمنيت الحصول على طريقة لصناعة تطبيقات لسطح المكتب للوندوز desktop applications بواسطة php التي احبها واحترفها ..

وقد عثرت على بعض البرامجيات في هذا الباب وكلها لم تكن تسرني ولا تفي بالغرض ...

فكنت اجرب من حين لحين لغات برمجية نصية تكون مفتوحة المصدر كالـ php فجرب البايثون والروبي وغيرها ووجدتها كلها بعيدة عن البساطة التي اجدتها في الـ php

وبعد ذلك عثرت على لغة بيرل perl ، فوجدتها قريبة تماما من php ولها نفس الاسلوب تقريبا ، فشرعت بتعلم اساسياتها ووجدتها كأنها نسخة معدلة من الـ php لسطح المكتب

فإذا كنت انت مبرمج بي اج بي فلن تجد صعوبة ابدا وبإمكانك البدء الان من خلال هذا الرابط

<https://www.tutorialspoint.com/perl/index.htm>

الان سنعرض كيفية انشاء واجهات المستخدم الرسومية GUI في لغة بيرل perl فلنبدأ على بركة الله .

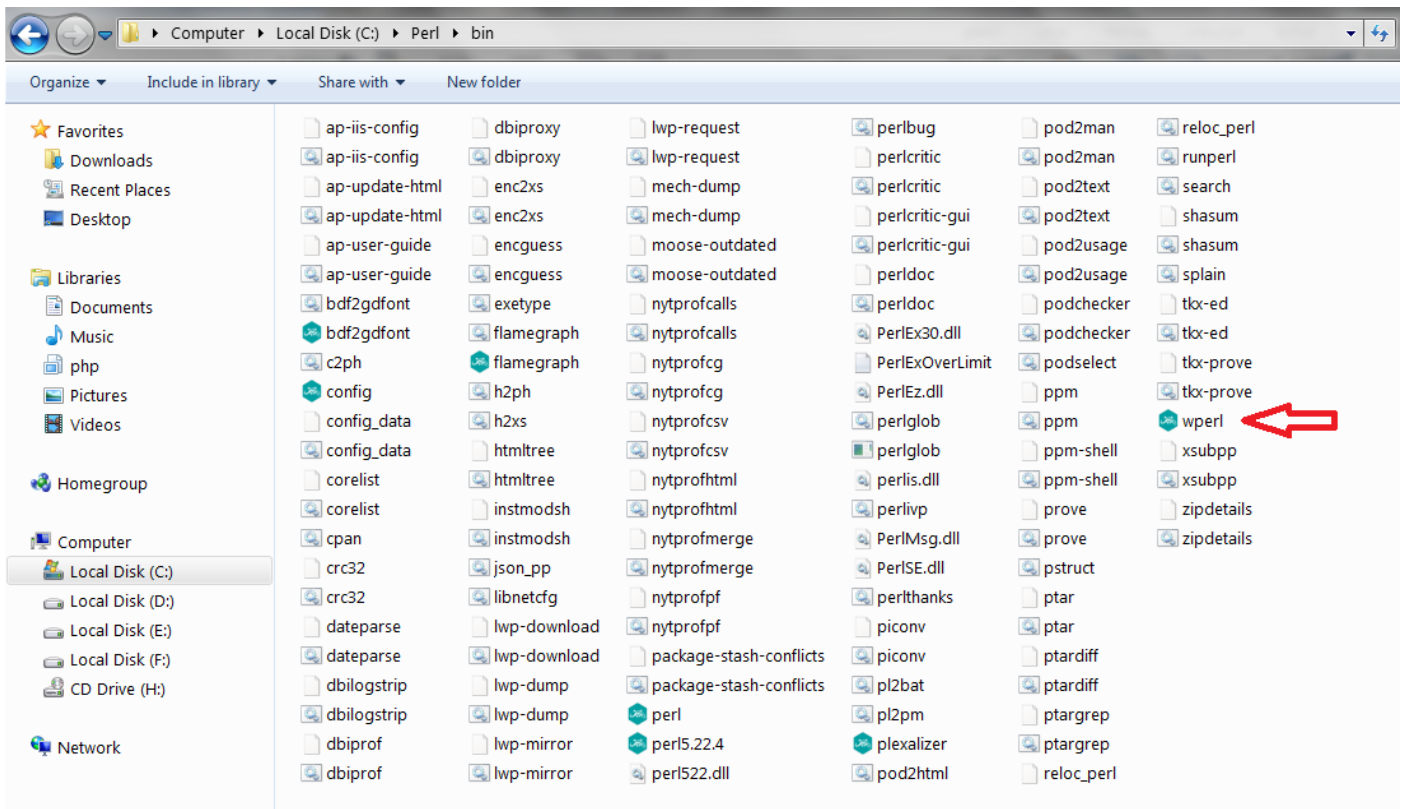
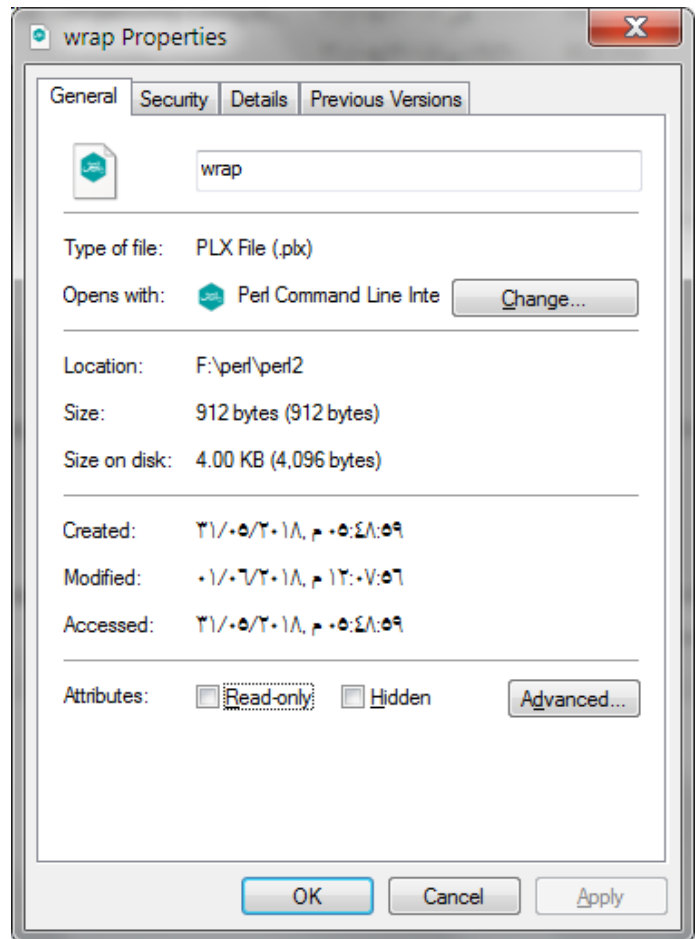
- قم بتنصيب لغة البيرل من موقع <https://www.activestate.com> وذلك لان نسخة هذا الموقع تحوي مكتبة مطورة من الـ tkx مدمجة مع اللغة.

بعد ذلك قم انشاء ملف نصي فارغ واحفظه بالامتداد .plx وليكن ملفك الاول مثلا perl1.plx

ستجد بعد ذلك نظام التشغيل الوندوز لم يتعرف على هذا النوع الجديد بينما هو يعرف النوع .pl

الفرق هو اننا لا نريد ان تظهر شاشة الكونسل السوداء مع الواجهات التي سنقوم بتصميمها لذلك اعمل رايت كلك على الملف ذي النوع .plx واختر الخصائص ومنها اختر open with واقرنه مع البرنامج الموجود بداخل مجلد لغة بيرل بالاسم wperl

وكما في الصور ادناه



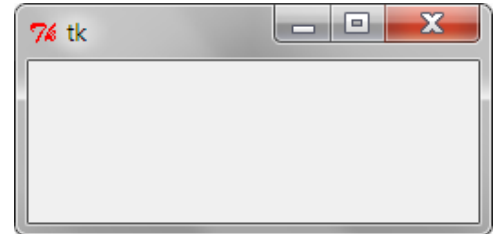
نستطيع الان عمل اول نافذة بواسطة الـ tkx في لغة البيزل وهي بهذا البساطة

my first comment line inside the gui window code. هذا سطر تعليق يبدأ بالهاش

```
use Tkx;
```

```
my $mw = Tkx::widget->new(".");
```

```
Tkx::MainLoop;
```



السطر الاول يقوم بادراج المكتبة الاحداث الخاصة بالواجهات الرسومية tkx

والحرفين tk اختصار لعبارة tool kit أي سبب ادوات

وهي مكتبة تستخدم مع البايثون والبيزل ولغات اخرى ، والنسخة الحالية الاحداث هي tkx

السطر الثاني يقوم بتعريف متغير على طريقة البيزل بواسطة كلمة my وهي المدى للمتغير كمتغير محلي .. وقد اخترت كما هو شائع استخدام الحرفين mw لمتغير النافذة لتقابلان عبارة main window أي النافذة الرئيسة وهي مبنية على كلاس tkx كما هو واضح ..

السطر الاخير يجعل النافذة تظهر وتنتظر استقبال الاحداث على مكوناتها (من ازرار مربعات نص وغيرها والاحداث هي من قبيل click , keypress, الخ) و (ستلاحظ ان أي خطأ في الكود يؤدي الى عدم ظهور النافذة)!

الان نريد اضافة بعض التحديدات على شكل النافذة فنضع لها العنوان اولا

والصيغة له تكون بهذا الشكل

```
$mw->g_wm_title("hello perl learners");
```

ولتحديد اقل حجم للنافذة نكتب هذا السطر

```
$mw->g_wm_minsize(500, 300);
```

فيكون مجموع السطور في البرنامج هكذا

```
use Tkx;
```

```
use utf8;
```

```
my $mw = Tkx::widget->new(".");
```

```
$mw->g_wm_title("hello perl learners");
```

```
$mw->g_wm_minsize(500, 300);
```

```
Tkx::MainLoop;
```

- اذا كنت ترغب في كتابة عنوان بالعربي للنافذة فعليك اولا ادراج مكتبة الترميز الخاصة في اعلى البرنامج وهي
عبارة: use utf8;

ادراج الليبل Label

لا تنسى انك تتعامل مع نماذج فلا تستطيع هنا الكتابة بشكل مباشر على النموذج كما تفعل مع صفحة الويب بل عليك ان تضع النصوص والارقام في وعاء خاص بها وهو اما أن يكون لاصقة ليبل او مربع نص او قائمة الخ

وهنا كيفية ادراج اللاصقة النصية label

```
my $label;
```

```
$label = $mw->new_label(-text=>"النص المطلوب عرضه");
```

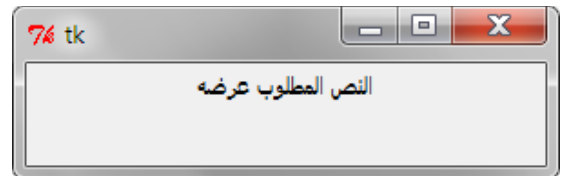
```
$label->g_pack();
```

السطر الاول قام بتعريف متغير خاص بالليبل وفي السطر الثاني تم تحديد مكان ظهور الليبل وهو النافذة الحالية \$mw

وتم تحديد خاصية النص لليبل بداخل الاقواس كما ترى اعلاه ، ولاحظ ان الخواص تبدأ بخط شارحة صغير يليها علامة ال يساوي والاكبر من ، والسبب ان ما تراه هو عنصر مصفوفة فيشار اليها كما تعلم بهذه الطريقة =>

السطر الاخير يقوم باظهار الليبل في المكان الذي نريده ، يعني ربما تأخر السطر الثاني وجاء في مكان اخر حيث يجب ان يظهر وفق التصميم الذي نريده ، وربما وضعناه بداخل حلقة تكرار وكلمة باك pack تعني رمز ، يعني ادفع للنافذة الليبل الذي وصفناه لك انفا ... والاقواس التي مع (دالة الباك) هي معدة لاستقبال محددات اخرى يوضحها المثال ادناه وهي المحددات التي تحدد كيفية و شكل ظهور الليبل والسطور اعلاه يمكن ان نكتبها في سطر واحد هكذا

```
my $label = $mw->new_label(-text=>"النص المطلوب عرضه")->g_pack();
```



وجرب ايضا اضافة مزيد من الخصائص هكذا

```
my $label = $mw->new_label(-text=>"النص المطلوب عرضه",
```

```
-font=>"Tahoma",
```

```
-background=>"red")->g_pack(-side=>"left");
```

ادراج حقل نص entry

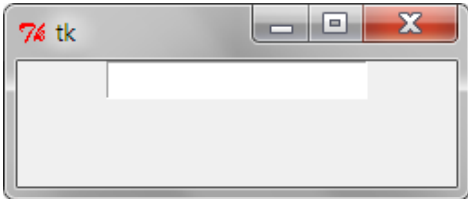
الانترى هو بمثابة textbox في الفجول بيسك مثلا ويمكن ادراجه بنفس الطريقة الخاصة بالليل هكذا

```
my $textbox = $mw->new_entry()->g_pack();
```

ولاجل تعيين قيمة للنص او التعامل من داخل الكود مع القيم التي سيقوم المستخدم لادخالها نحتاج الى انشاء متغير وتعيينه كمتغير خاص بالانترى بهذه الطريقة

my \$content

```
my $textbox = $mw->new_entry(-textvariable=>\$content)->g_pack();
```



ادراج نص طويل text

وهو يختلف عن حقل النص اعلاه من حيث كونه اطول ويمكن ادراج اكثر من سطر فيه ويكتب هكذا

```
my $longtext = $mw->new_text(-font=>"arial 14 {bold}");
```

```
$longtext->insert("end","Go to Sleep\n");
```

```
$longtext->g_pack();
```

حيث يقوم السطر الاول بتعريف اشارة او مقبض الى الكائن النصي من خلال تعريف متغير بالاسم \$longtext

ثم ينشئ النص في النافذة الحالية \$mw وبين الاقواس يعني له الخط الافتراضي

السطر التالي يقوم بادراج نص اولي للمربع النص هذا

وبعد ذلك يمكن رزم الكائن لاطهاره على النموذج ، ولن يظهر مربع النص اذا قمنا بالرزم pack() قبل ادراج النص الابتدائي ، واما اذا اردنا ان يظهر مربع النص فارغا بدون نص ابتدائي فيمكن الاستغناء عن السطر الثاني ويكون الكود بهذه الصورة

```
my $longtext = $mw->new_text(-font=>"arial 14 {bold}")->g_pack();
```

الكود بهذا الشكل الاساسي سوف ينتج مربع نص يملأ جميع المتبقي من مساحة النافذة ، اما اذا اردنا تقليصه فيمكن تحديد العرض والارتفاع له محسوبا بعدد الاسطر

```
my $longtext = $mw->new_text(
```

```
-font=>"arial 14 {bold}",
```

```
-width => 40, -height => 10,
```

```
)->g_pack();
```

ولمربع النص الطويل هذا ايضا صفة او متغير الـ `textvariable` - الذي يجري من خلاله التحكم في محتوى الكائن عبر الكود ..

اذن هكذا رأينا ان الصيغة العامة لادراج كائن `widget` هي ان نضع متغيرا نسنده اليه مقبض هذا الكائن ونقوم بتحديد مكان الكائن ونوع الكائن ووضع الصفات له ثم الرزم لظهور الكائن على النافذة

```
My $thing = $mainwindowname->new_widgetType->(-property=>value,..)->g_pack();
```

بهذا سيظهر الكائن على النافذة او الفورم ، ولكن المتغير الخاص به سوف ينفذ لاحقا في التعديل على الكائن خلال الكود وقراءة واسناد القيم له ...

زر الامر Button widget

```
my $btn = $mw->new_button(-text=>"click me",-command=>\&dothis)->g_pack();
```

هذا هو كود توليد زر الامر في لغة البيزل ، وتلاحظ وجود كلمة كوماندا -command وهي خاصية من خلالها نسمي اسم الاجراء الفرعي الذي سيتم تنفيذه عند الضغط على هذا الزر ، وهنا انا اعطيت له الاسم dothis وسنكتبه ادناه

```
Sub dothis {mkdir "NewFolderFromPerlCode";}
```

وكما تلاحظ ايضا فان اسم الاجر مسبق بالشرطة المائلة باك سبيس \ ورمز & والشرطة المائلة هي حرف هروب بينما & هي التي تجعل لغة بيرل تتعرف على كلمة dothis على انها اسم اجراء وليس نصا عاديا فقط ، على كل حال اكتبها كما هي ..

وكما ترى وتلاحظ فان سطر الكود بداخل هذا الاجراء الفرعي سيقوم بعمل مجلد جديد بالاسم NewFolderFromPerlCode في المسار الحالي الذي يوجد فيه ملف السكريبت الحالي الذي نقوم بتنفيذه.



وهناك طريقة اخرى لكتابة نفس الكود اعلاه اقصد زر الامر+الاجراء التابع له ك بلوك واحد بهذه الصورة

```
my $btn = $mw->new_button(-text=>"click me",-command=>sub {  
mkdir "NewFolderFromPerlCode";  
})->g_pack();
```

اكتفي بهذا المقدار على امل انك عرفت الان كيفية ادراج الادوات ضمن النموذج وتستطيع القراءة من المصدر الرسمي للموقع الخاص بال- tkx في هذا الرابط <https://tkdocs.com/tutorial/index.html>

#Messages from perl

```
use Tkx;

use utf8;

my $mw = Tkx::widget->new(".");

$mw->g_wm_minsize(400,300);

$mw->g_wm_maxsize(400,300);

#=====

my $ent = $mw->new_entry(-textvariable=>\$msgbox)->g_pack();

$mw->new_button(-text=>"show box content",-command=>\&dothis)->g_pack();

$mw->new_button(-text=>"show message",-command=>\&msg)->g_pack();

#=====

Tkx::MainLoop();

sub dothis {

if ($msgbox){

Tkx::tk__messageBox(-message=>$msgbox,-title=>"hello");}

else {Tkx::tk__messageBox(-message=>"type something",-title=>"hello");}}

sub msg {Tkx::tk__messageBox(-message=>"hello again",-title=>"hello");

$mw->configure(-bg=>'#ffff00');

}
```

text-format

```
use Tkx;

use utf8;

my $mw = Tkx::widget->new(".");

my $longtext = $mw->new_text(
    -font=>"Arial 16 {bold}",
    -background=>"wheat",
    -foreground=>"blue",
    #-height=>15,
    #-width=>20,
    -wrap=>"word",
    -selectbackground=>"blueviolet",
    -setgrid=>true
)->g_pack();

Tkx::MainLoop();
```

directory files

```
use Tkx;
```

```
use utf8;
```

```
my $mw = Tkx::widget->new(".");
```

```
my $itm;
```

```
my $lb = $mw->new_listbox(-selectbackground =>"blue",-borderwidth=> 3,-  
listvariable=>$itm);
```

```
#اكتب اسم لمجلد اخر عندك بدلا من الحالي في هذا المثال
```

```
$dirname = "/perl";
```

```
opendir ( DIR, $dirname ) || die "Error in opening dir $dirname\n";
```

```
while( ($filename = readdir(DIR)) ) { $lb->insert("end", "filename : " . $filename);}
```

```
closedir(DIR);
```

```
$lb->g_pack(-fill=>"both");
```

```
Tkx::MainLoop();
```

listbox + for loop + scrollbar

```
use Tkx;

my $mw = Tkx::widget->new(".");

($lb = $mw->new_tk__listbox(-height => 5))->g_grid(-column => 0, -row => 0, -sticky =>
"nwes");

($s = $mw->new_ttk__scrollbar(-command => [$lb, "yview"],
    -orient => "vertical"))->g_grid(-column =>1, -row => 0, -sticky => "ns");

$lb->configure(-yscrollcommand => [$s, "set"]);

($mw->new_ttk__label(-text => "Status message here",
    -anchor => "w"))->g_grid(-column => 0, -row => 1, -sticky => "we");

($mw->new_ttk__sizegrip)->g_grid(-column => 1, -row => 1, -sticky => "se");

$mw->g_grid_columnconfigure(0, -weight => 1); $mw->g_grid_rowconfigure(0, -weight =>
1);

for ($i=0; $i<10; $i++) {

    $lb->insert("end", "Line " . $i . " of 100");

}

Tkx::MainLoop();
```

two froms

```
use Tkx;

use utf8;

my $mw = Tkx::widget->new('.');

$mw->g_wm_title("Hello, world");

    $mw->g_wm_minsize(500, 300);

my $btn = $mw->new_button(-text=>"open window",-command=>\&newform)->g_pack();

my $lbltxt = $mw->new_label(-text=>"Enter name :")->g_pack(-anchor=>"nw",-side=>"left");

    my $txt = $mw->new_entry()->g_pack(-anchor=>"nw",-side=>"left");

    my $lbltxt2 = $mw->new_label(-text=>"Enter school name :")->g_pack(-anchor=>"nw");

    my $txt2 = $mw->new_entry()->g_pack(-anchor=>"nw",-side=>"left");

Tkx::MainLoop;

#-----

sub newform {

    $w = $mw->new_toplevel;

    $w->g_wm_title("Hello, again");

        $w->g_wm_minsize(200, 300);

    my $btn = $w->new_button(-text=>"hello??",-command=>\&hello)->g_pack();

    my $lbl = $w->new_label(

        -text=>"اهلا بك في النموذج الاخر",

        -font=>"Tahoma 12 {bold}",

        -relief=>"ridge",

        -bg=>"gold")->g_pack(-side=>"top",-fill=>"both"); }

#-----

sub hello { $w->g_wm_title("Yes Hello, again"); }
```

```
# more formats

use strict;

use Tkx;

use utf8;

my $mw = Tkx::widget->new('.');

$mw->g_wm_title("Hello, world");

    $mw->g_wm_minsize(500, 300);

        my $content = "نص بداخل الصندوق";

my $lbl = $mw->new_label(
-text=>"اول واجهة يبدو الامر سهلا جدا",
-font=>"Tahoma 12 {bold}",-relief=>"ridge",-bg=>"gold")->g_pack(-side=>"top",-fill=>"both");

my $ent = $mw->new_entry(-textvariable=>\$content,-font=>"Tahoma",-borderwidth => 2,-
relief=>"raised")->g_pack(-side=>"top");

my $btn2 = $mw->new_button(-text=>"exit",-command=>sub{$mw->g_destroy();})->g_pack();

my $chkvar="on";

my $chkbtn = $mw->new_checkbutton(-text=>"male",-variable=>\$chkvar)->g_pack();

my $rdbtn = $mw->new_radiobutton(-text=>"ready",-variable=>\$chkvar)->g_pack();

my $skl = $mw->new_scale(-from=>1,-to=>50,-orient=>"horizontal",-label=>"درجة الحرارة")-
>g_pack(-fill=>"x",-expand=>1);

#my $txt = $mw->new_text(-font=>"Arial",-background => "teal",->g_pack(-pady=>12);

Tkx::MainLoop();
```

dialogs

```
use strict;

use Tkx;

use utf8;

my $mw = Tkx::widget->new('.');

$mw->g_wm_title("Hello, world");

    $mw->g_wm_minsize(500, 300);

        my $content;

my $ent = $mw->new_entry(-textvariable=>\$content,-font=>"Tahoma",-borderwidth => 3)->g_pack();

my $opn = $mw->new_button(-text=>"open file",-command=>sub {

my $filename = Tkx::tk___getOpenFile();

$content = $filename;

})->g_pack(-side=>"left");

my $opn = $mw->new_button(-text=>"save as",-command=>sub {

my $filename = Tkx::tk___getSaveFile();

$content = $filename;

})->g_pack(-side=>"left");

#-----

my $dirbtn= $mw->new_button(-text=>"directory",-command=>sub {

my $dirname = Tkx::tk___chooseDirectory();

$content = $dirname ;

})->g_pack(-side=>"left");

#-----

my $opn = $mw->new_button(-text=>"colors",-command=>sub {

my $color = Tkx::tk___chooseColor(-initialcolor => "#ff0000");

$mw->configure(-background=>$color);

})->g_pack(-side=>"left");

Tkx::MainLoop();
```

run a program

```
use Tkx;

my $mw = Tkx::widget->new(".");

$mw->g_wm_title("Hello, run a program from here");

$mw->g_wm_minsize(300, 200);

my $bt = $mw->new_button(-text=>"click me",-command=>sub {exec "calc.exe";})->g_pack();

Tkx::MainLoop();
```

open text file

في هذا المثال عليك توفير ملف نصي بجوار البرنامج وبالإسم file.txt

```
use strict;

use Tkx;

use utf8;

my $mw = Tkx::widget->new(".");

    $mw->g_wm_title("titlo capes");

$mw->g_wm_geometry("1000x600-200+10");

my $longtext = $mw->new_text(-font=>"arial 14 {bold}");

open(DATA, "<file.txt") or die "Couldn't open file file.txt, $!";

while(<DATA>) {

    $longtext->m_insert('end', "$_");

}

$longtext->g_pack();

Tkx::MainLoop();
```

واخر دعوانا ان الحمد لله رب العالمين

اسماعيل خليل ابو باسل البصرة ٢-٦-٢٠١٨

abubasil70@gmail.com