

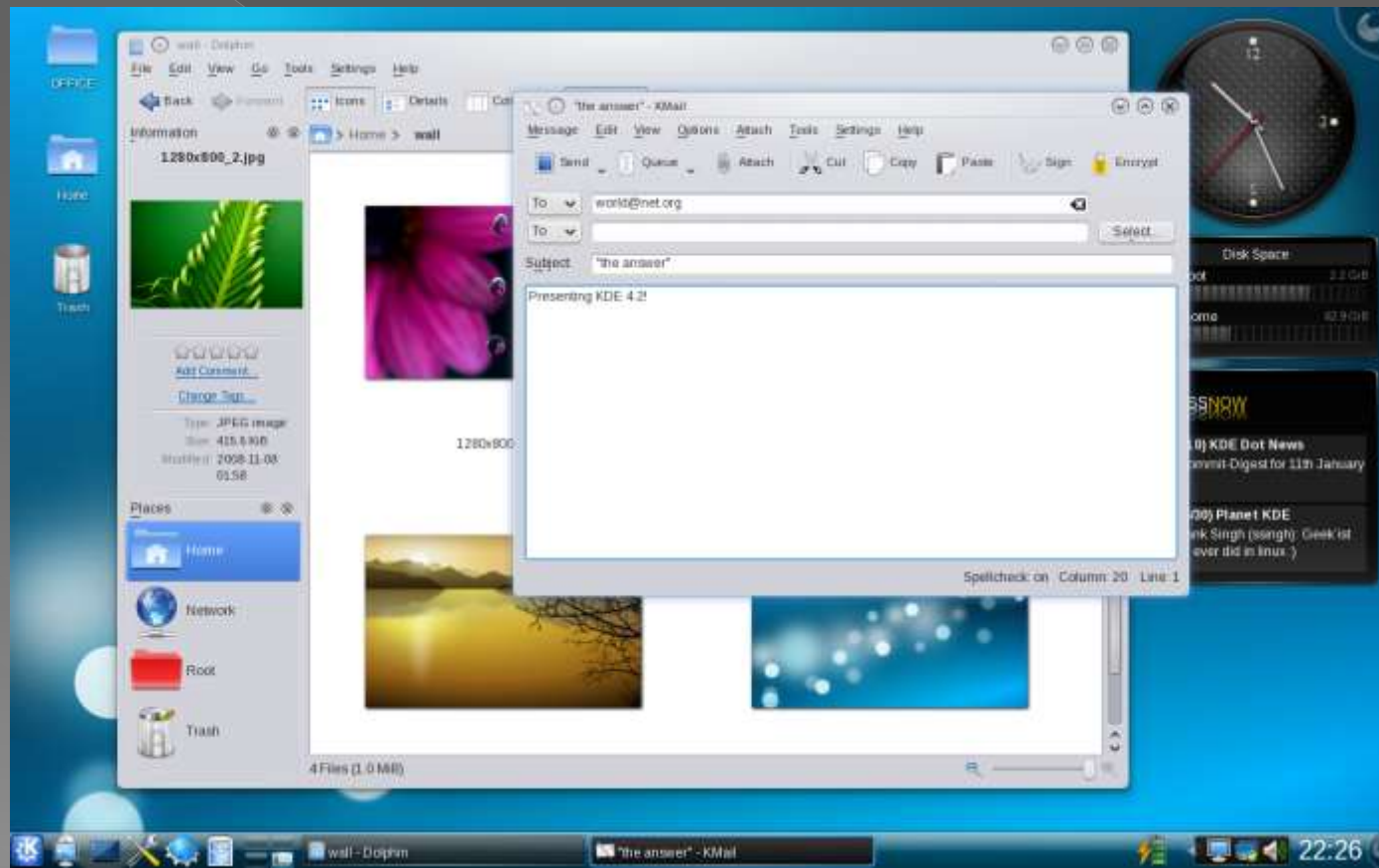
QT 入門介紹

資工105 游輝亮

QT

- ◎ 一個跨平台的原生C++應用程式開發框架
- ◎ 自由且開放原始碼的軟體
- ◎ 最早為 Trolltech 公司所有，後為 Nokia併購
- ◎ Qt使用標準的C++和特殊的代碼生成擴充功能（稱為元物件編譯器（Meta Object Compiler, moc））以及一些巨集。通過語言綁定，其他的程式語言也可以使用Qt。
 - Fortran QT
 - Python QT

使用 QT 製作的軟體



跨平台特性

- 使用Qt開發的軟體，相同的程式碼可以在任何支援的平台上編譯與執行，而不需要修改原始碼。會自動依平台的不同，表現平台特有的圖形介面風格
 - > Linux/X11：支援 Kdevelop 和 Eclipse IDE 整合
 - > MAC：基於 Cocoa 框架
 - > Embedded Linux
 - > Windows
 - > Android
 - > IOS
- 「Code Less; Create More; Deploy Anywhere.」

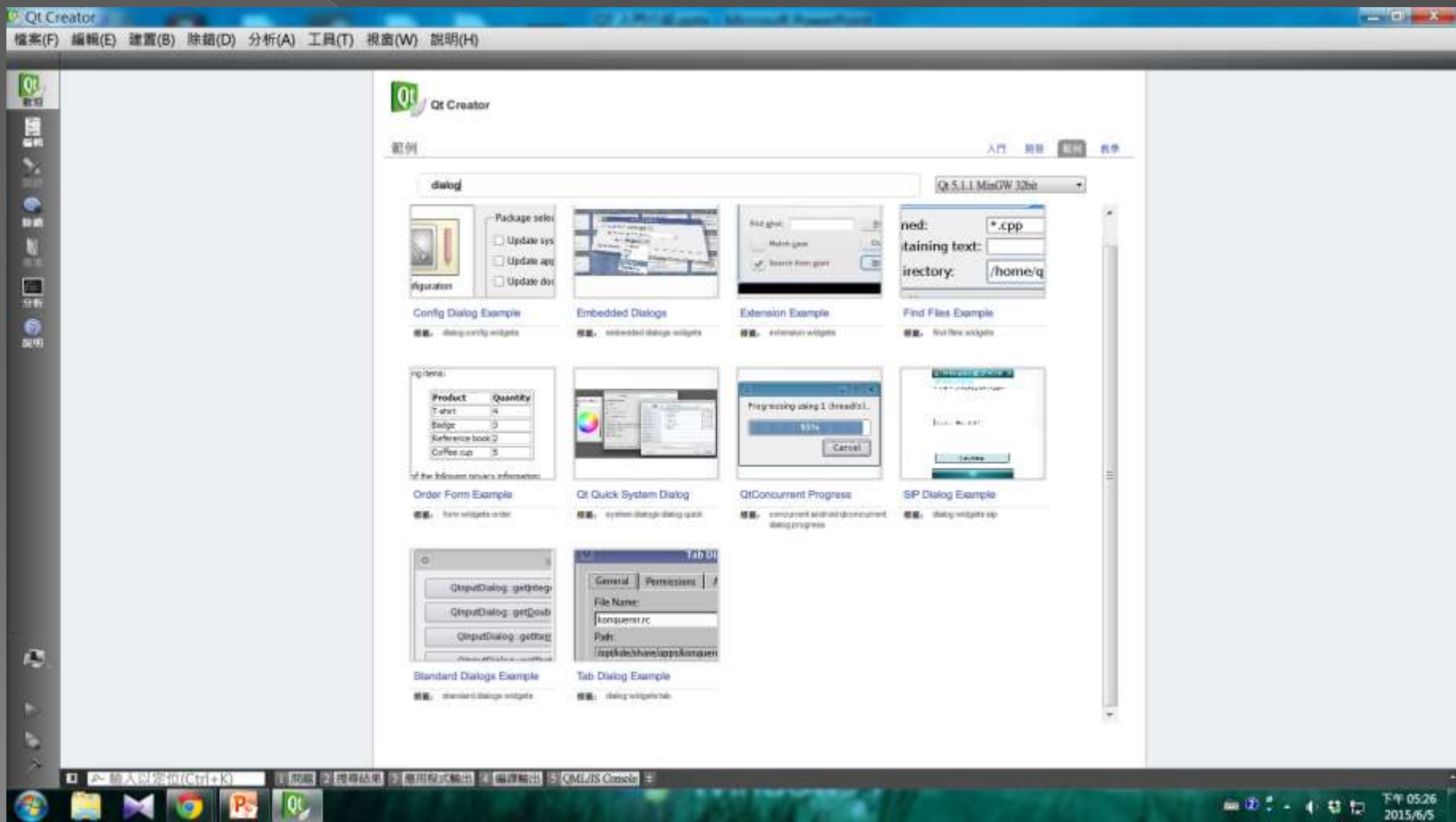
QT Library

- ◎ **QtCore**：所有基於 Qt 的應用程式的基礎，提供信號與槽的物件間通訊機制、IO、多執行緒
- ◎ **QtGui**：開發圖形使用者介面應用程式所需的功能，支援各個平台的原生圖形API，也支援反鋸齒、向量形變。
- ◎ **QtMultimedia**：多媒體內容處理
 - 存取攝影機及音頻功能的API
- ◎ **QtNetwork**：支援通用協定
 - HTTP、FTP和 DNS，與較低層的 TCP/IP 和 UDP 協議

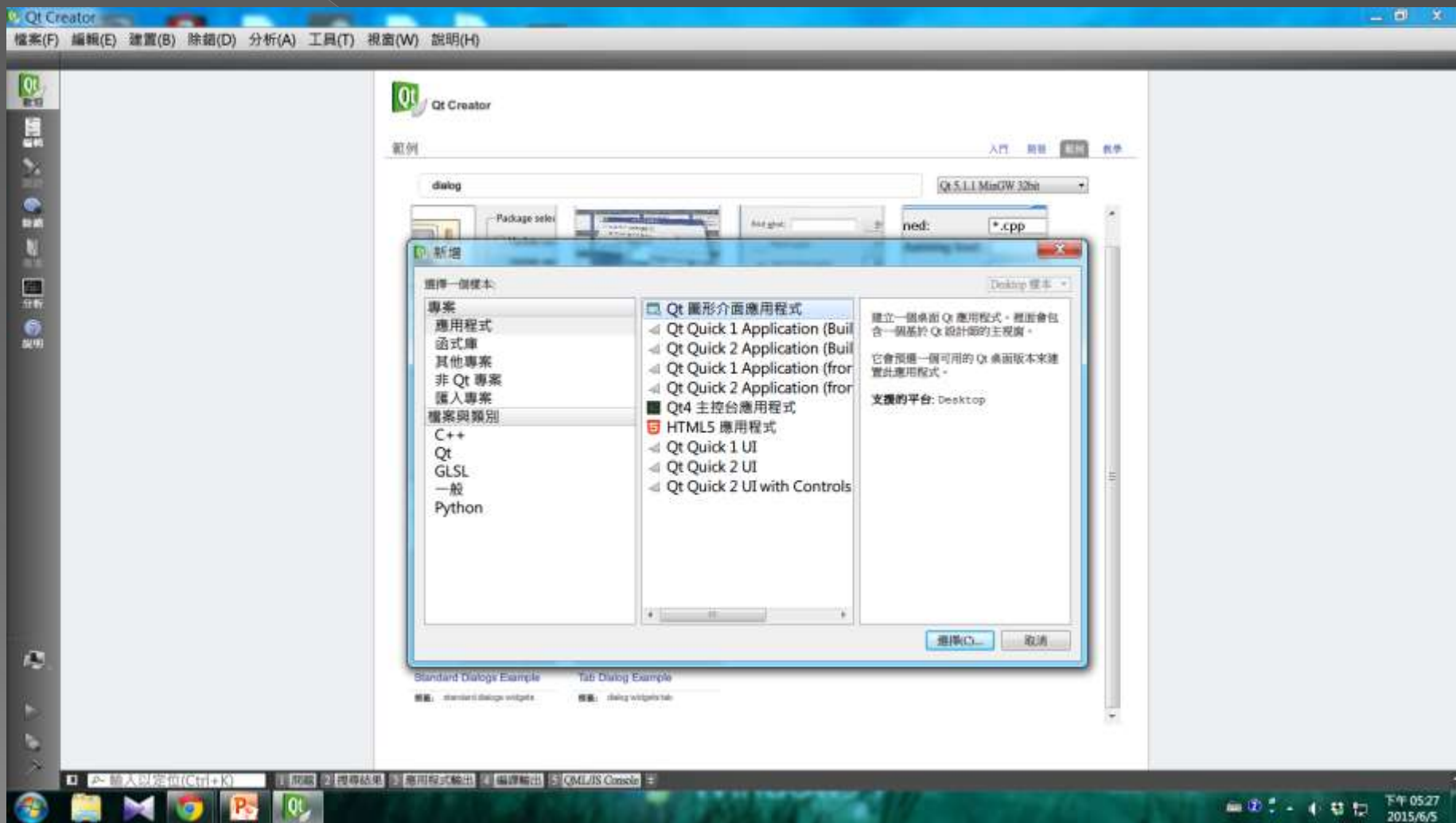
QT Library

- ◉ QtScript：完全整合的ECMA標準指令碼引擎
- ◉ QSql：支援所有主要的資料庫驅動
 - ODBC、MySQL、PSQL、SQLite、ibase、Oracle、Sybase、DB2
- ◉ QtWebKit：提供了HTML瀏覽器引擎
- ◉ QtXml：提供了XML文件的閱讀器和編寫器

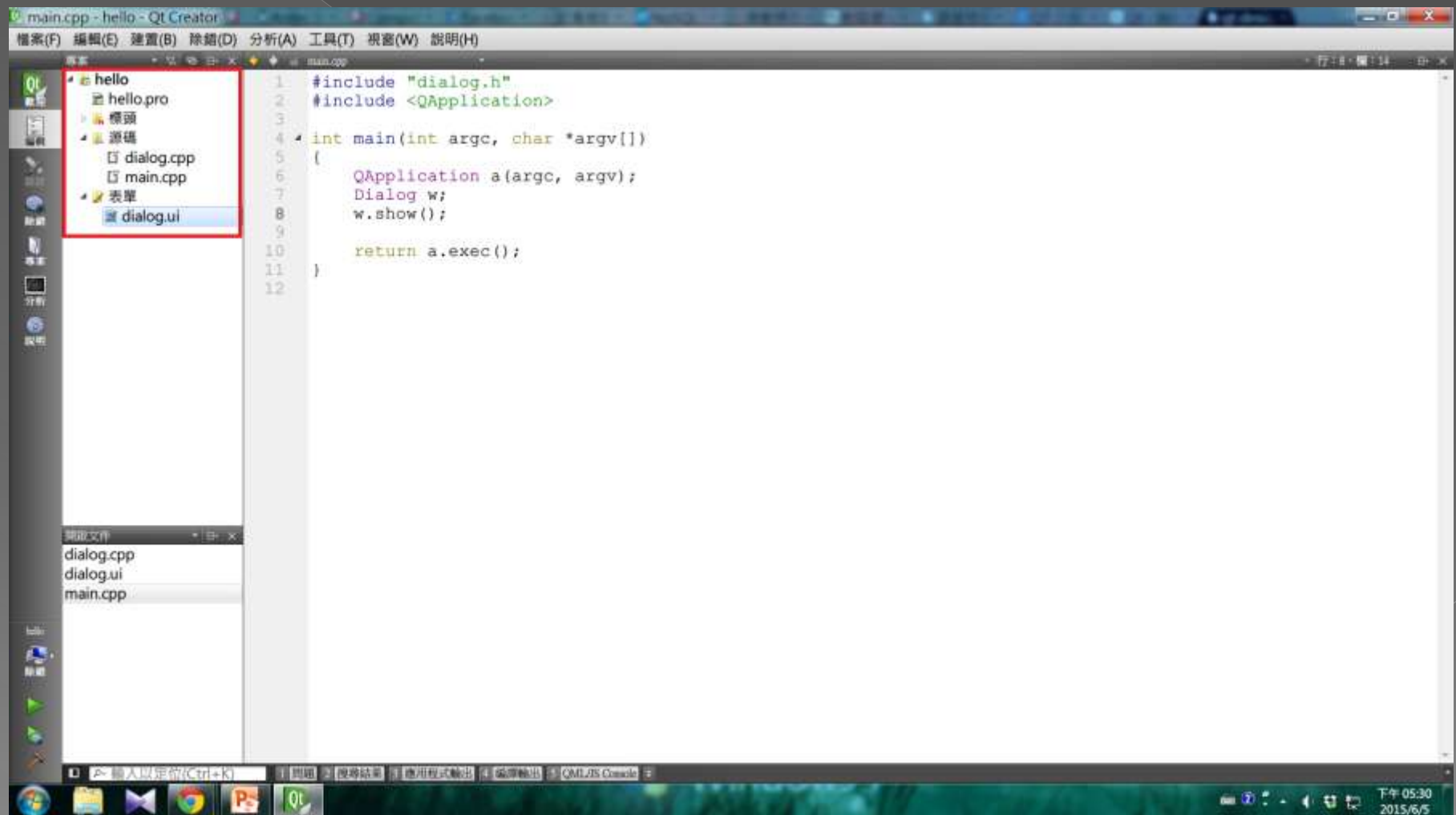
Hello world !!



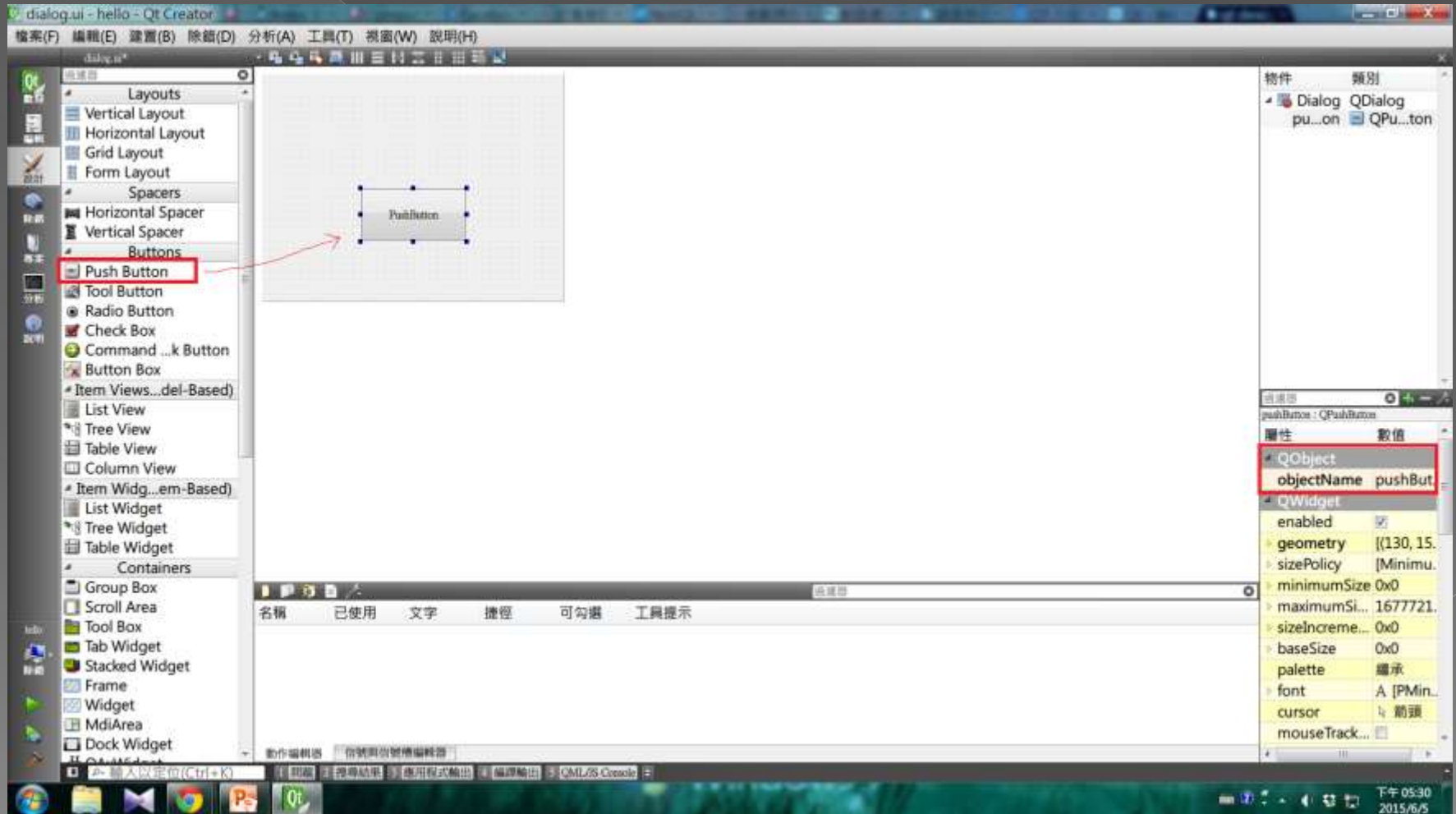
Hello world !!



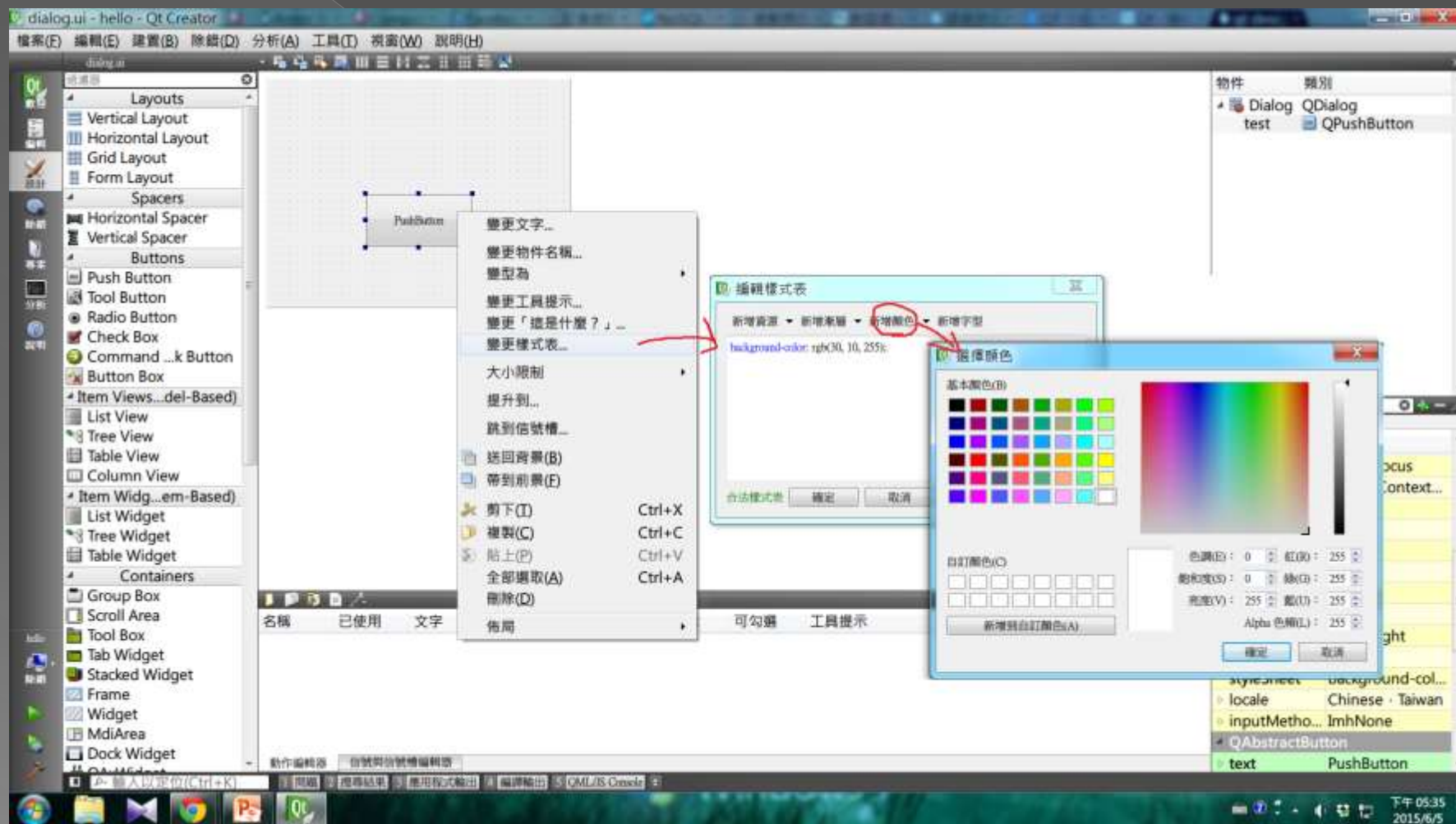
Hello world !!



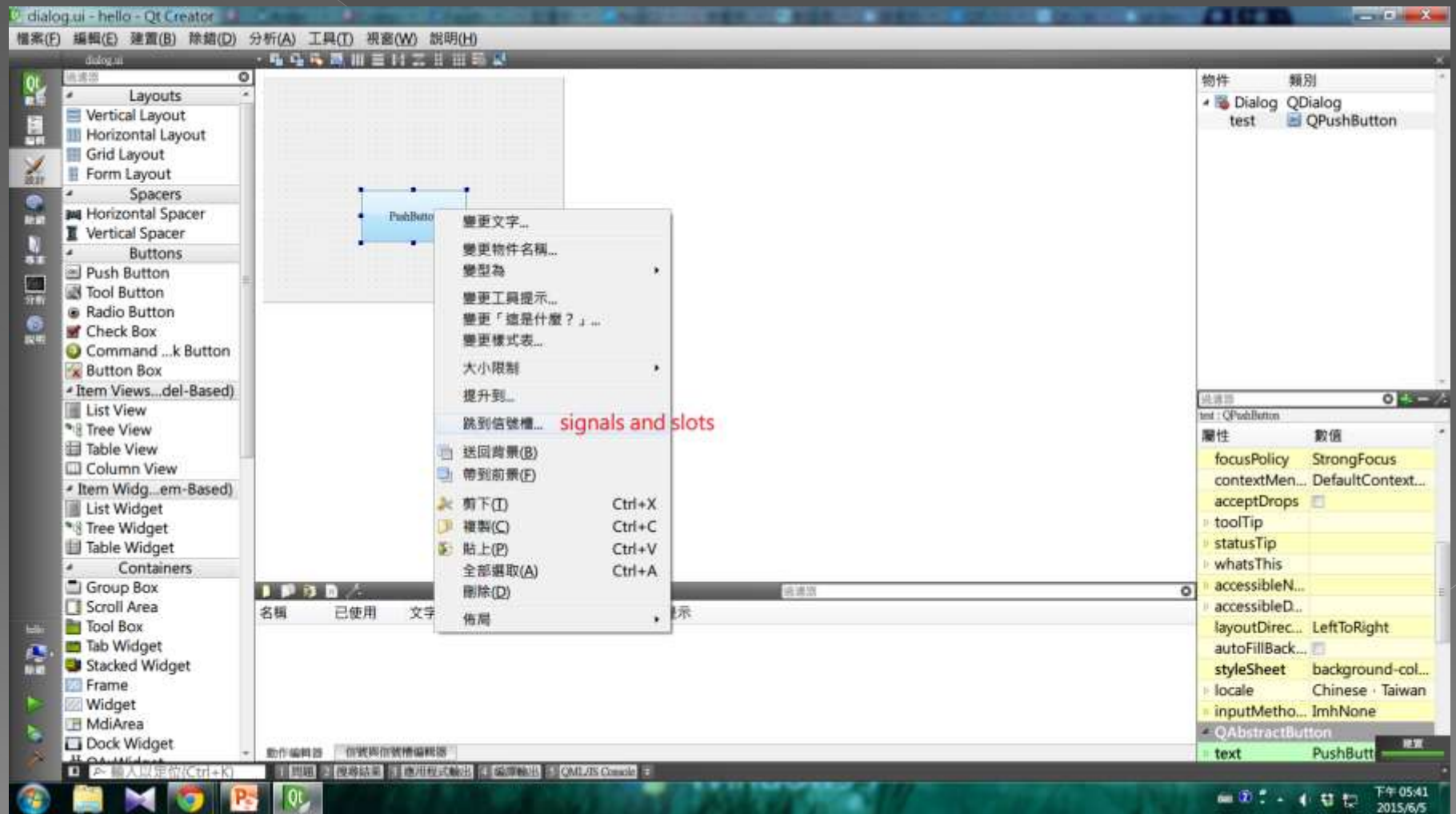
Hello world !!



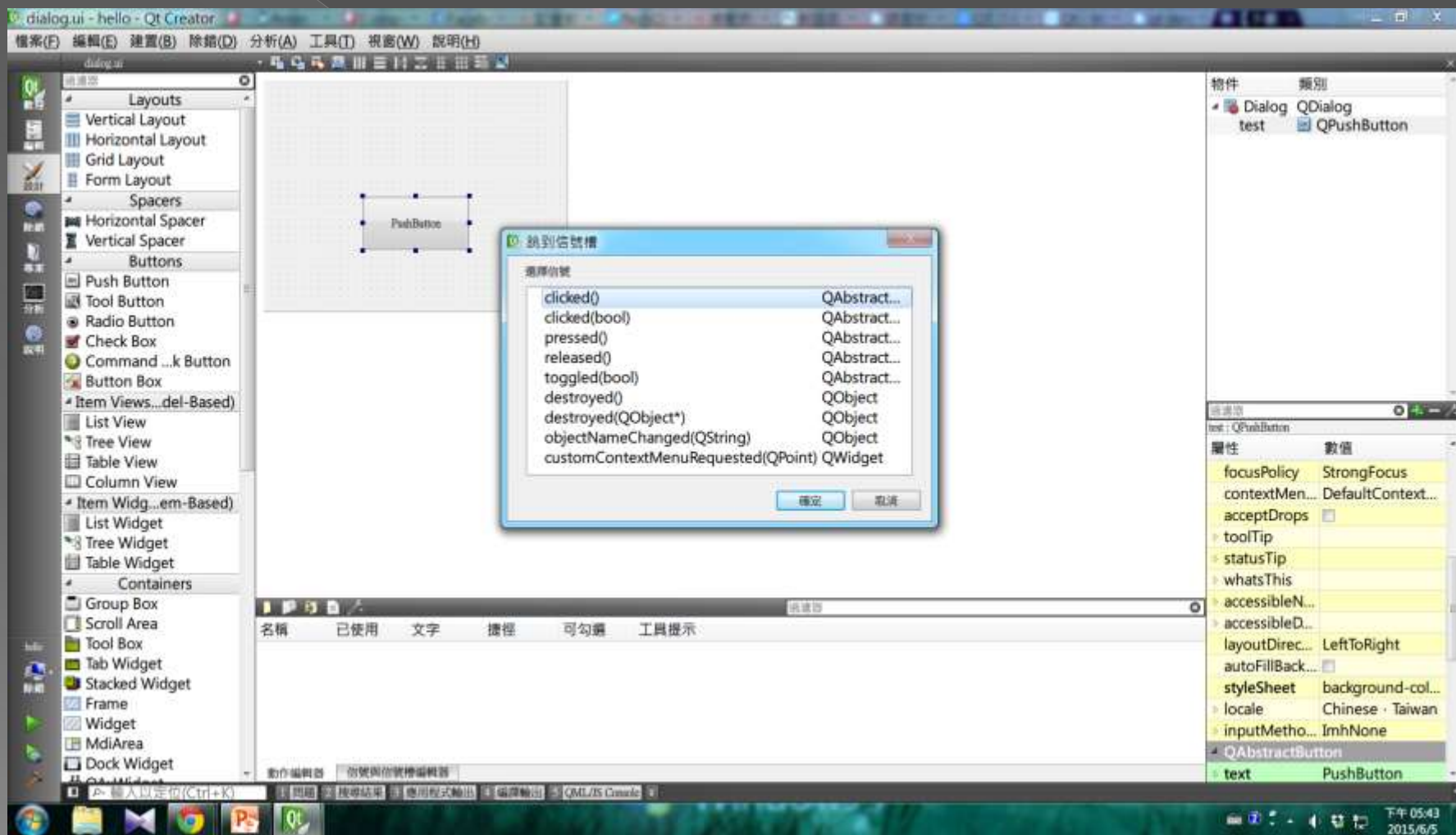
Hello world !!



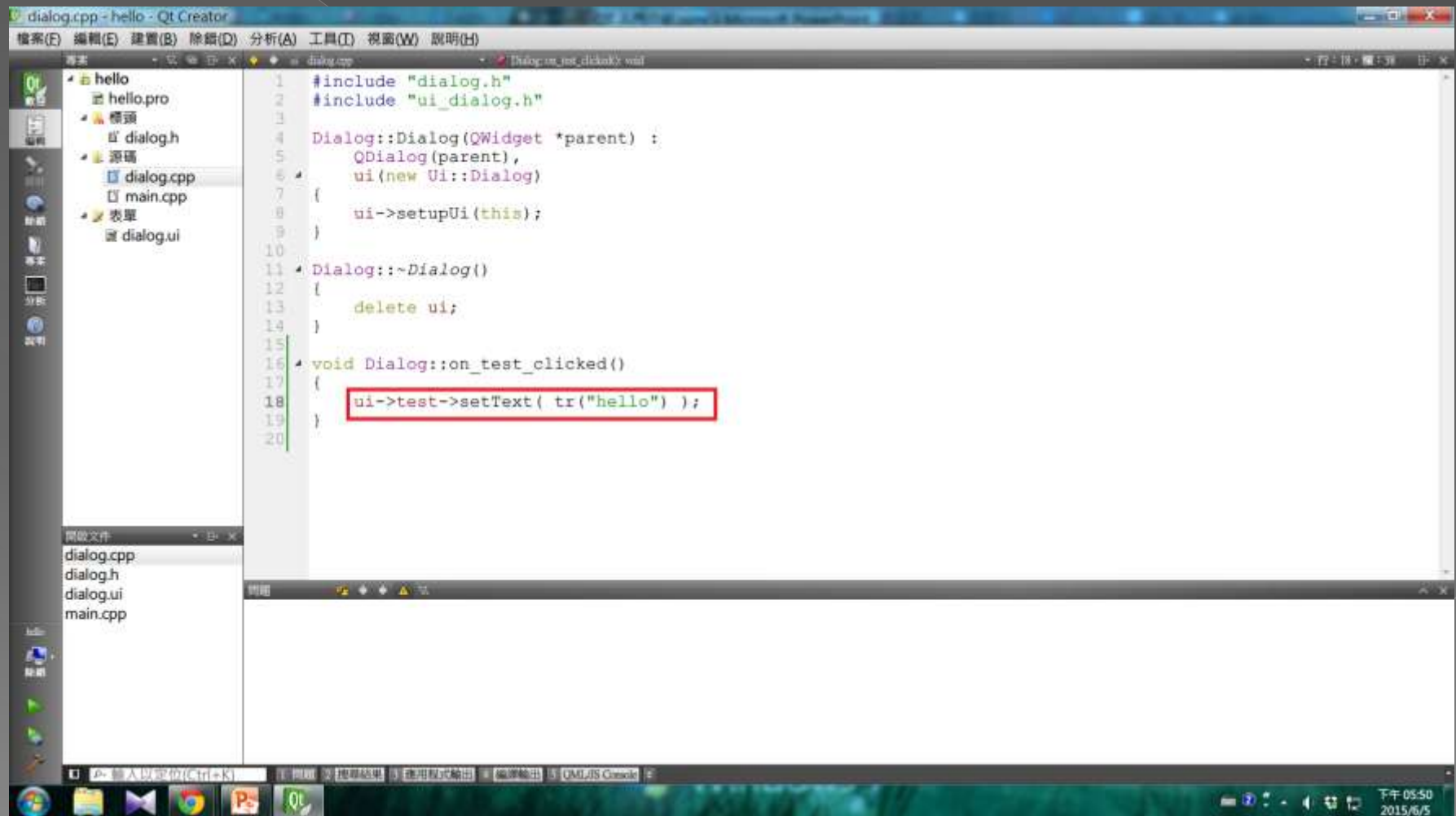
Hello world !!



Hello world !!



Hello world !!



Hello world !!



```
// Main message loop:
MSG msg;
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```



Signal and slot

Signal and slot

- Qt利用訊號與槽（signals/slots）機製取代傳統的 callback 來進行物件之間的溝通。當操作事件發生的時候，物件會發送出一個訊號（signal）；而槽（slot）則是一個函式接受特定信號並且執行槽本身設定的動作。訊號與槽之間，則透過 QObject 的靜態方法connect來連結。
- 訊號在任何執行點上皆可發射，甚至可以在槽裡再發射另一個訊號，訊號與槽的連結不限定為一對一的連結，一個訊號可以連結到多個槽或多個訊號連結到同一個槽，甚至訊號也可連接到訊號。

Signal and slot

- 以往的 callback 缺乏類型安全，在呼叫處理函式時，無法確定是傳遞正確型態的參數。但訊號和其接受的槽之間傳遞的資料型態必須要相符合，否則編譯器會提出警告。訊號和槽可接受任何數量、任何型態的參數，所以訊號與槽機制是完全類型安全。
- 訊號與槽機制也確保了低耦合性，發送訊號的類別並不知道是哪個槽會接受，也就是說一個訊號可以呼叫所有可用的槽。此機制會確保當在"連接"訊號和槽時，槽會接受訊號的參數並且正確執行。

Signal and slot 簡易版解釋

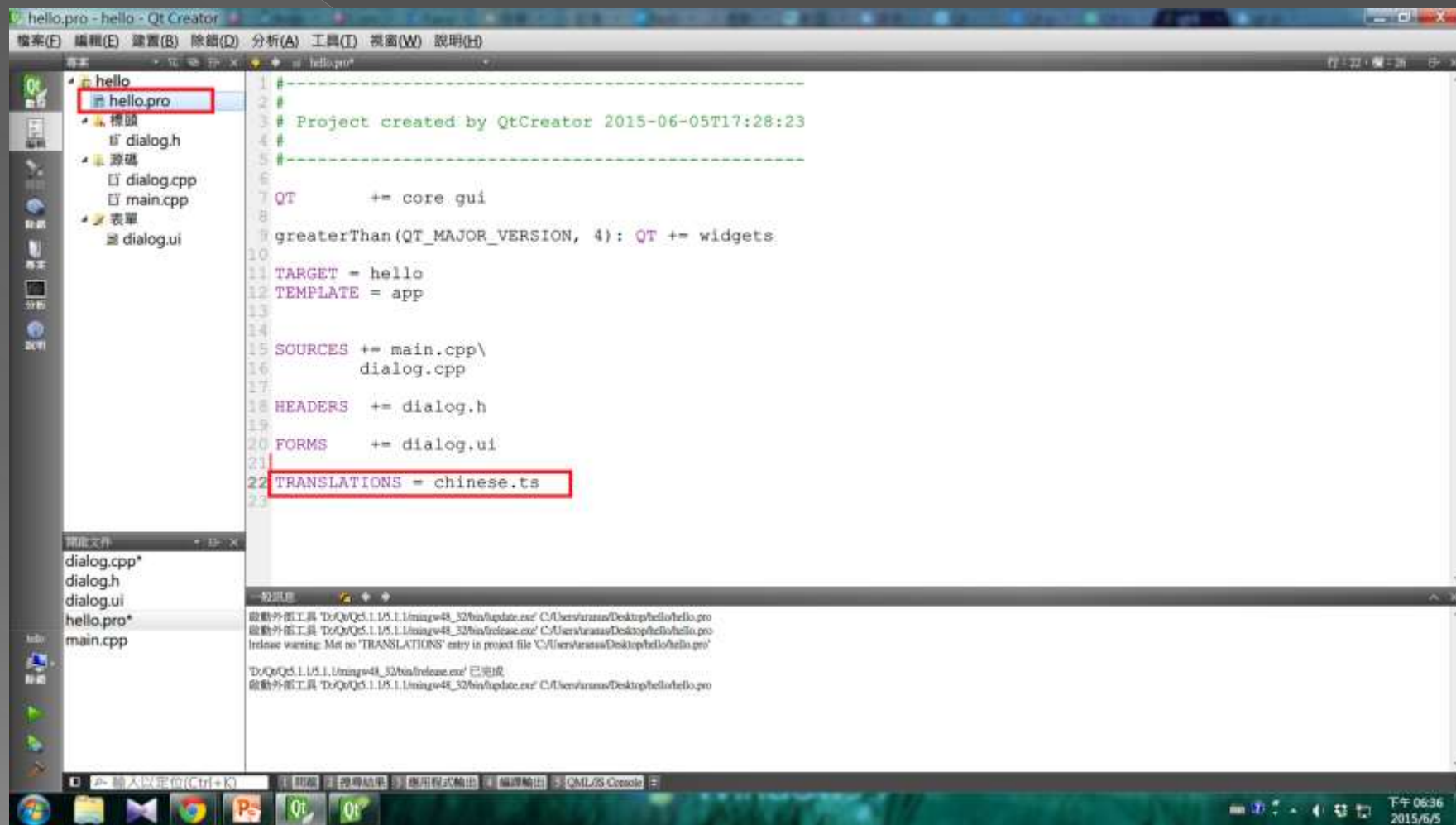
- 有一位 L 跑去找 R，然後他們兩個一見面就抱在一起開始放閃，完全不管別人。有些人呢，很興奮的叫好；有些人躲起來；另一些人則是想扁人。
- 就 Qt 系統來說，L 和 R 的行為放出了一個 signal；而對此回應的行為叫 slot。所以，「叫好」是 slot，「躲起來」是 slot，「扁人」也是 slot。
- 槽可以想成 signal 插入 slot，進而觸發行為。如上面的例子，L 和 R 的行為放出一個 signal，而這 signal 正是放閃光。
- 閃光最後「插」進每個人的眼中...等一下！slot 不是眼睛嗎？
- OK，眼睛收到 signal 後，根據它的 slot 又放出一個 signal 給大腦。大腦經過判斷後，總算放出 signal 給身體其它部位的 slot...
- 所以我們有如此格式：
- `connect(送出signal的人, SIGNAL(signal()), 收到signal的人, SLOT(slot()));`
- 最後我們寫出：
- `connect(L和R, SIGNAL(閃光()), 某A, SLOT(叫好()));`
- `connect(L和R, SIGNAL(閃光()), 某B, SLOT(躲起來()));`
- `connect(L和R, SIGNAL(閃光()), 某C, SLOT(扁人()));`

國際化佈署

tr("string") 是什麼？

- 讓 Qt 的字體引擎能夠在同一時間正確的顯示各種不同的書寫系統。並且使用 Qt 內部 Unicode 編碼來儲存文字。
- Text / code independent
- Qt的多國語言支援技術，可以讓應用程式中的文字全部使用英文撰寫，能夠在完全不需修改程式的狀況下，改變整個應用程式中的文字為另一個語系的文字，並能夠協助處理不同語言的單、複數問題。

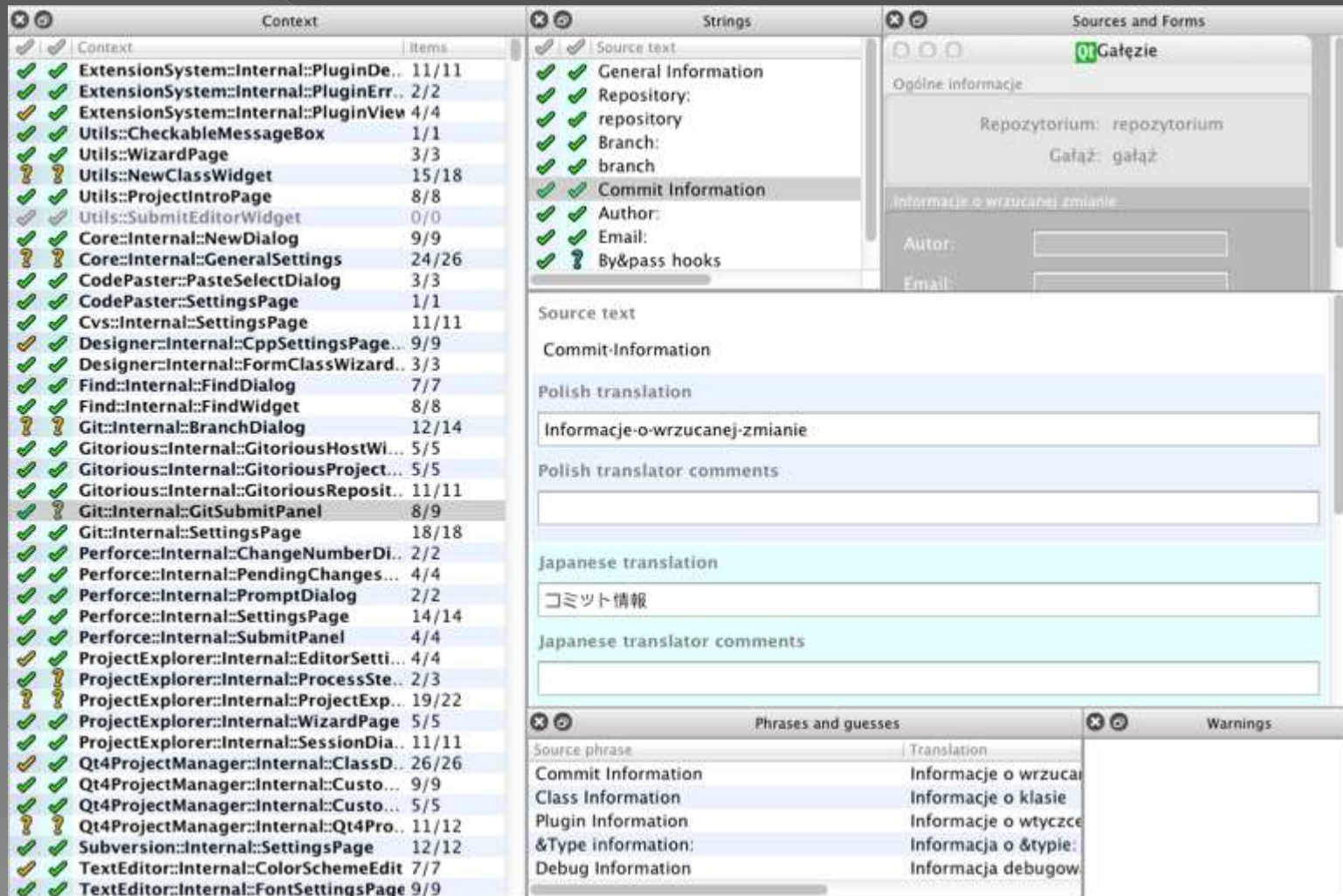
tr("string") 是什麼?



tr("string") 是什麼?



tr("string") 是什麼?



tr("string") 是什麼?

- ◉ When you finish it, release it to get "qm" file.
- ◉ When you want to translate your application, load the "qm" file.

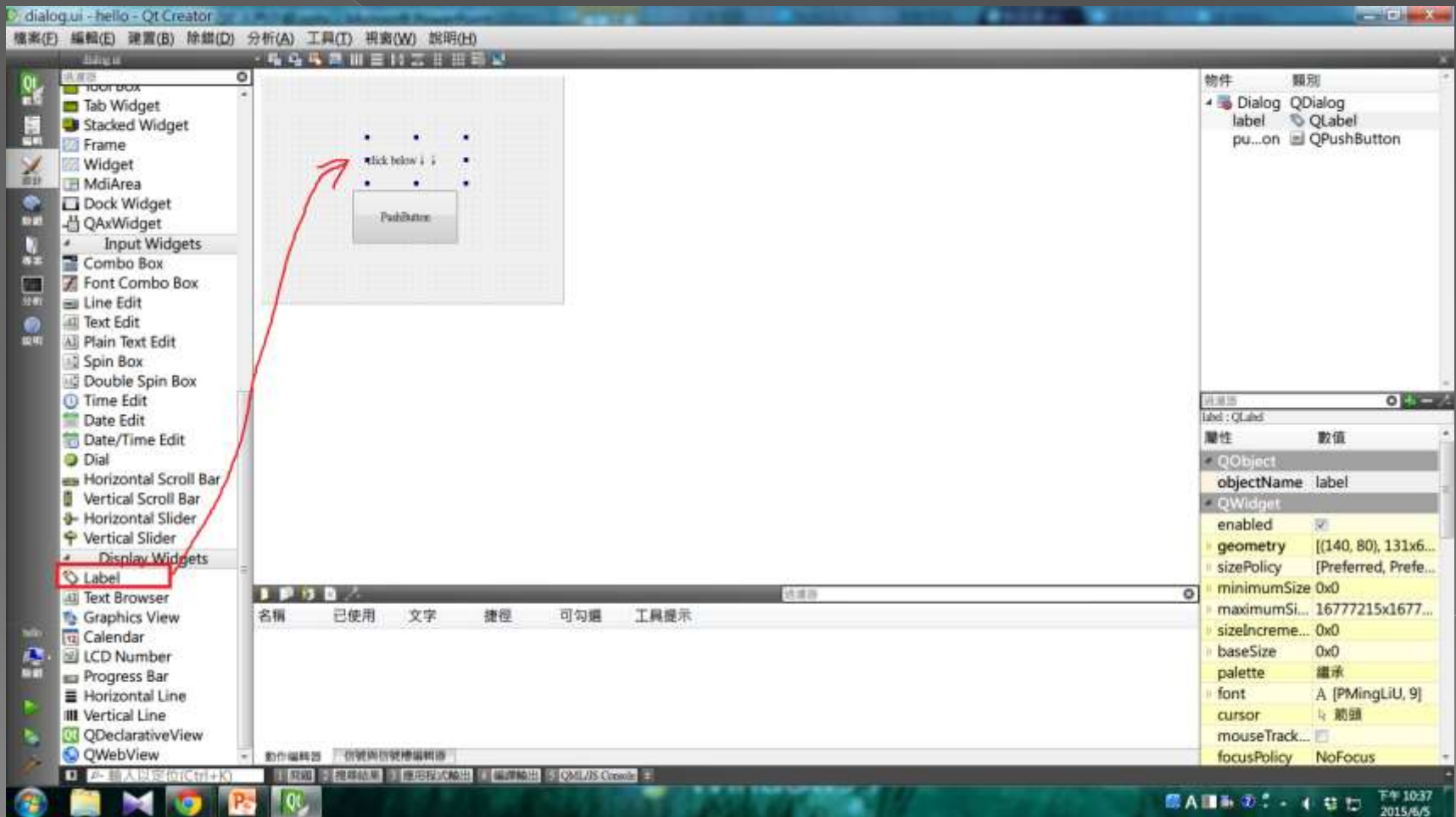
```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);

    QTranslator *translator = new QTranslator;
    translator->load("chinese.qm");

    a.installTranslator(translator);
}
```

Message Box

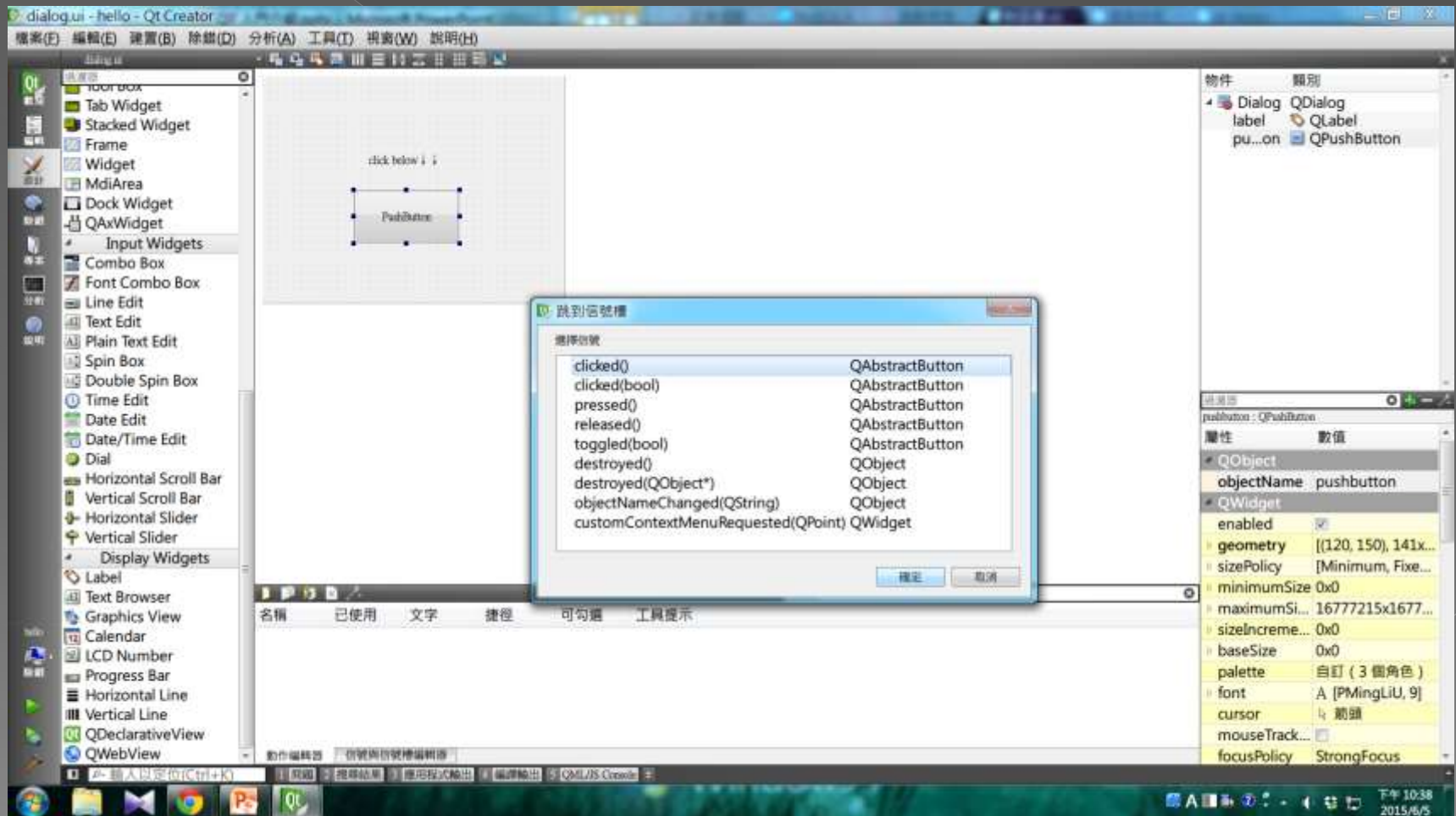
Message Box



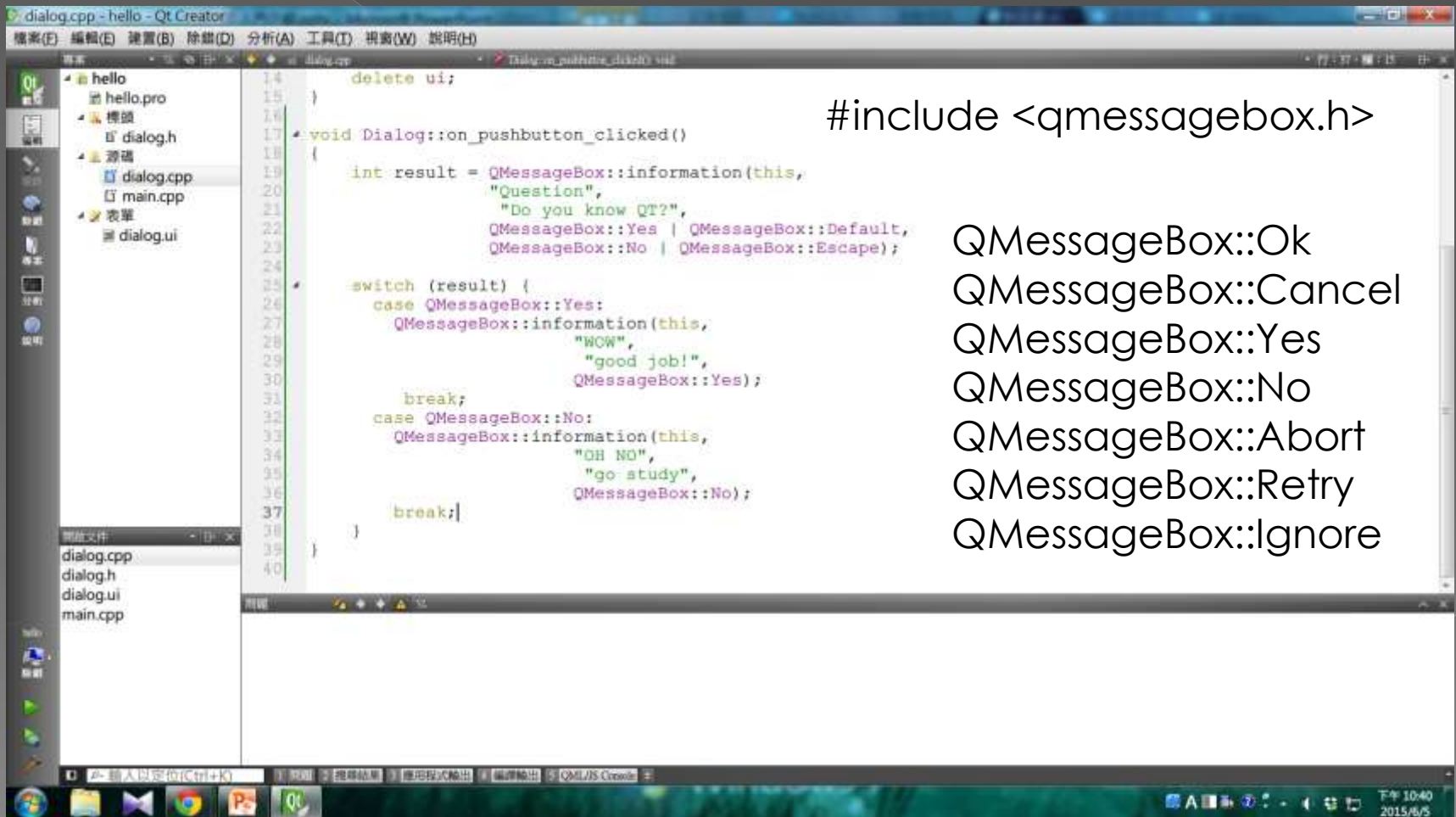
Message Box



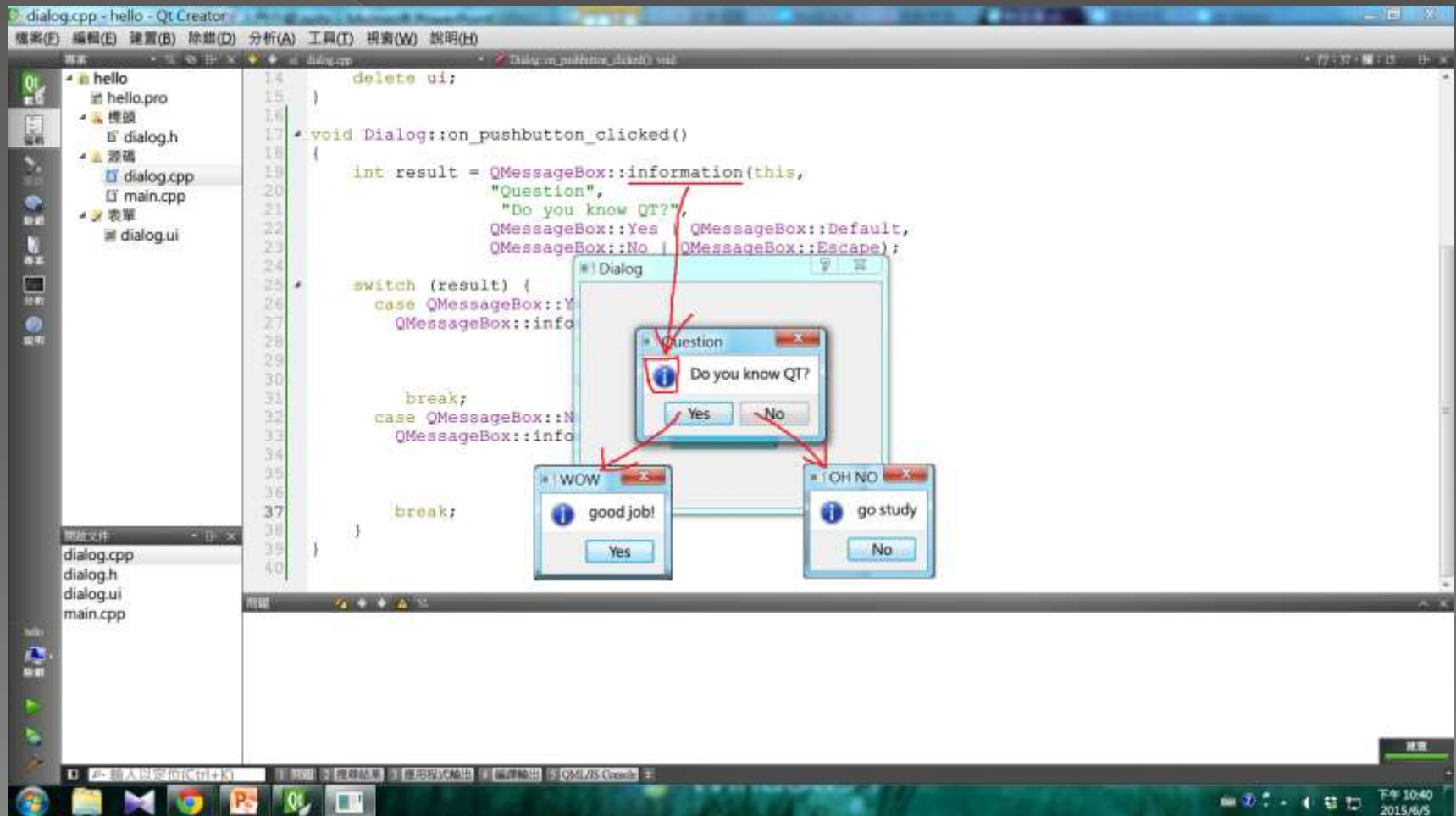
Message Box



Message Box



Message Box



Message Box

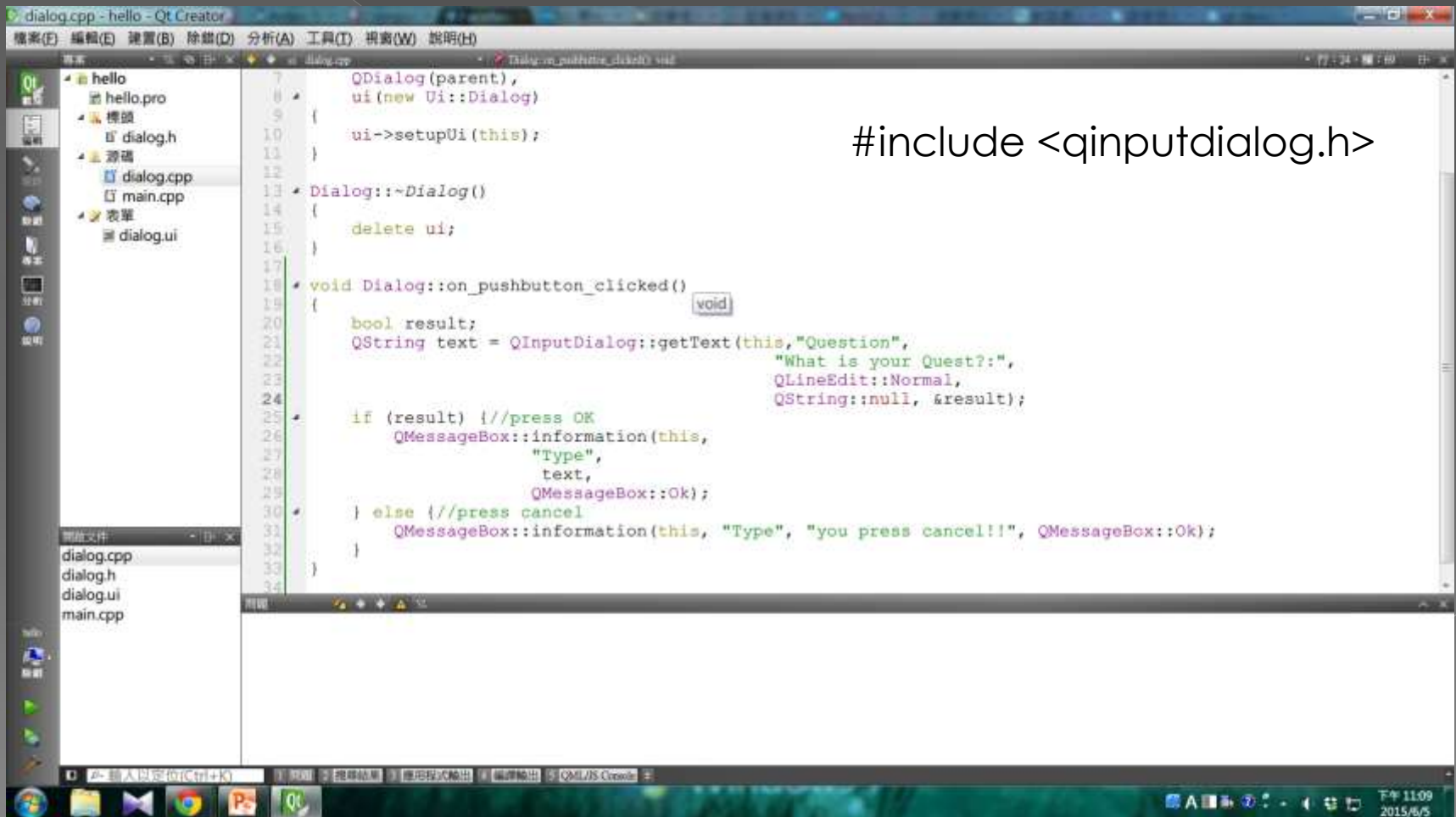
- QMessageBox可以使用一些基本的 html 語法來設定文字的顯示，訊息方塊將顯示 Yes 與 No 兩個按鈕，在組合時的列舉值是 StandButton 列舉 (enum) 值，可以參考 [線上文件的表格](#) 進行對照。



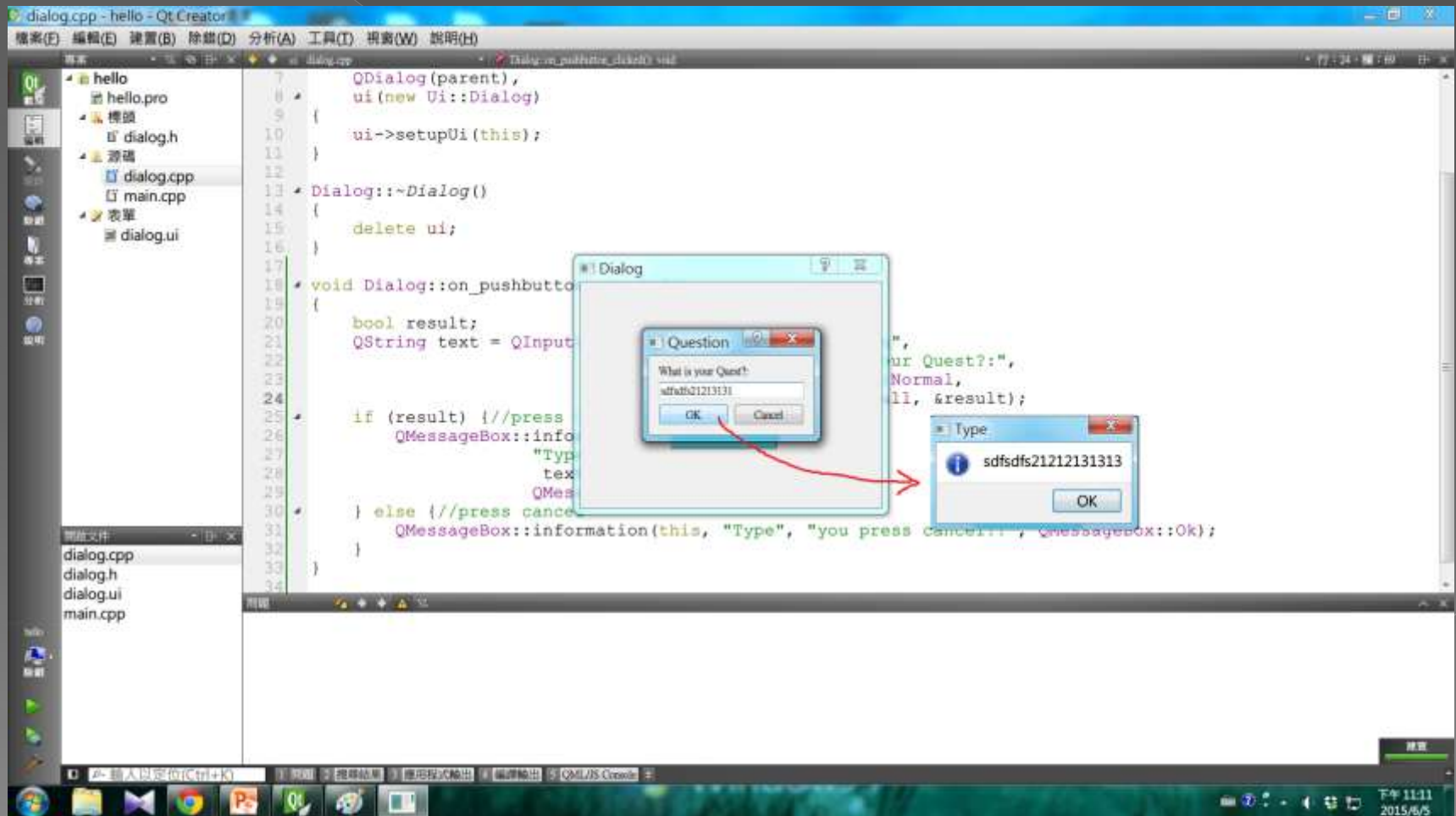
Input Dialog



Input Dialog



Input Dialog



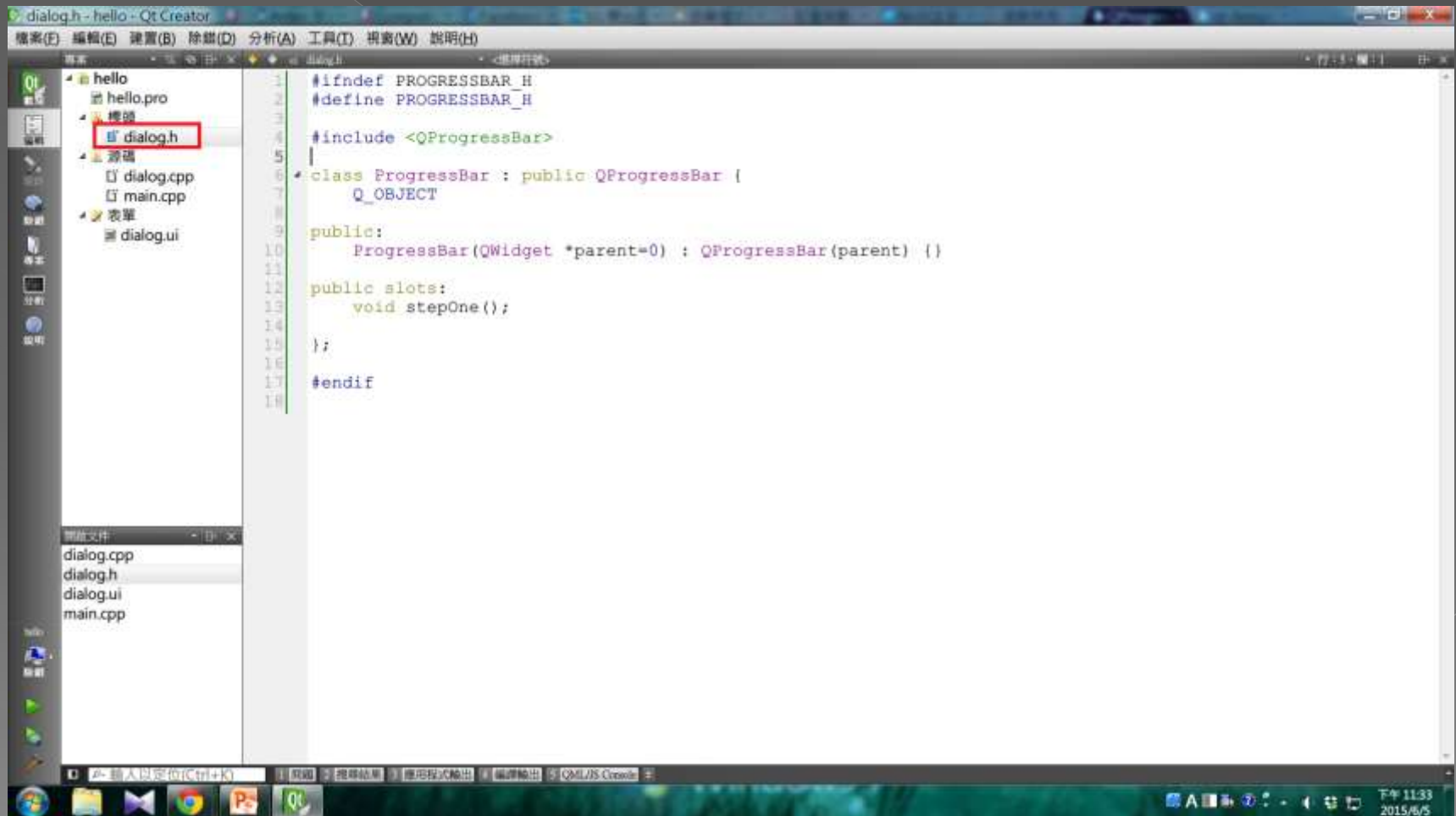
Input Dialog

- 設定預設值為0，下界為0，上界為10，小數位數為2位的浮點數輸入對話方塊：
- `double input = InputDialog::getDouble(parent, "Input Dialog", "Enter a double", 0.0, 0.0, 10.0, 2, &isOK);`

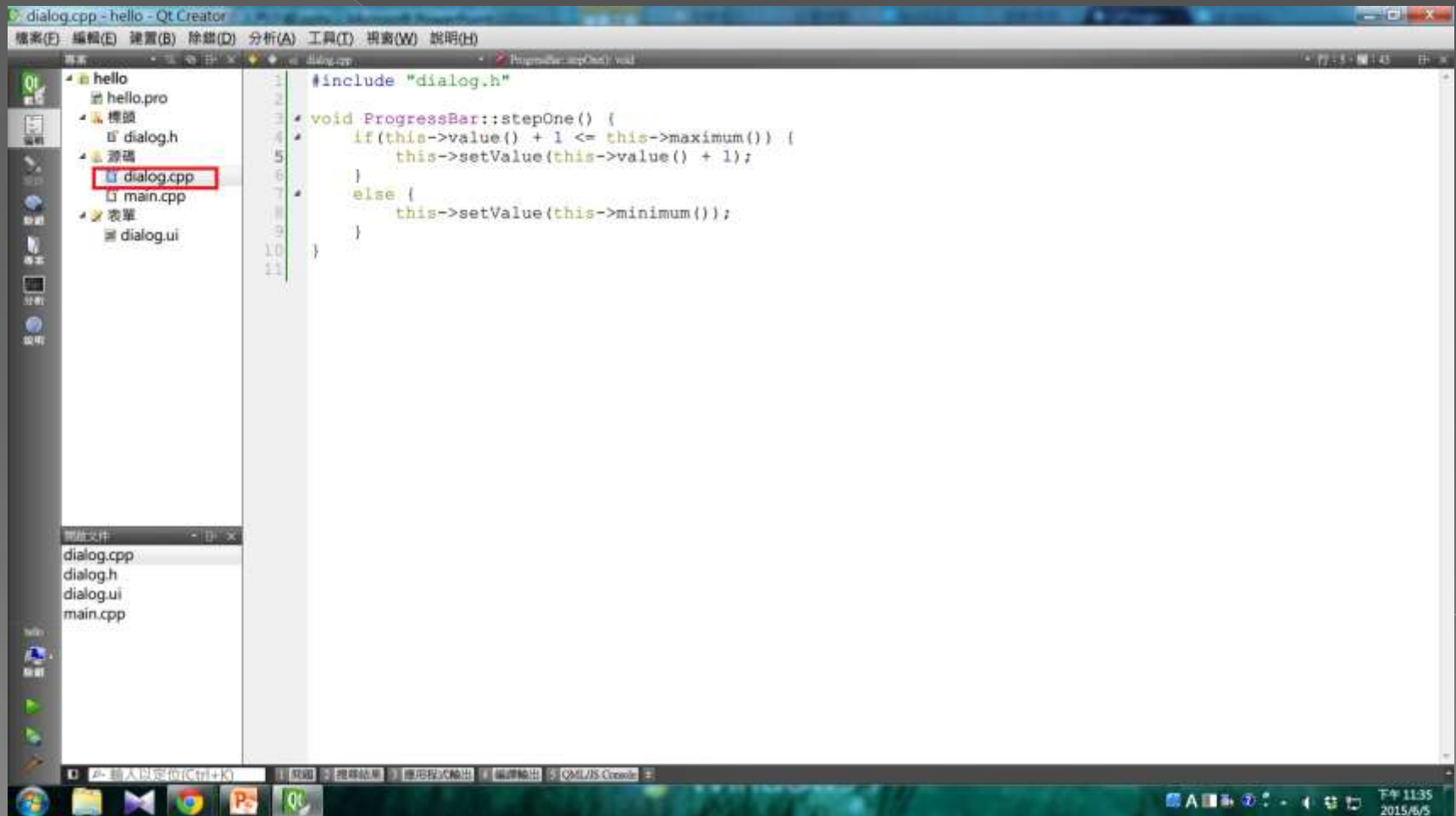


Progress Bar

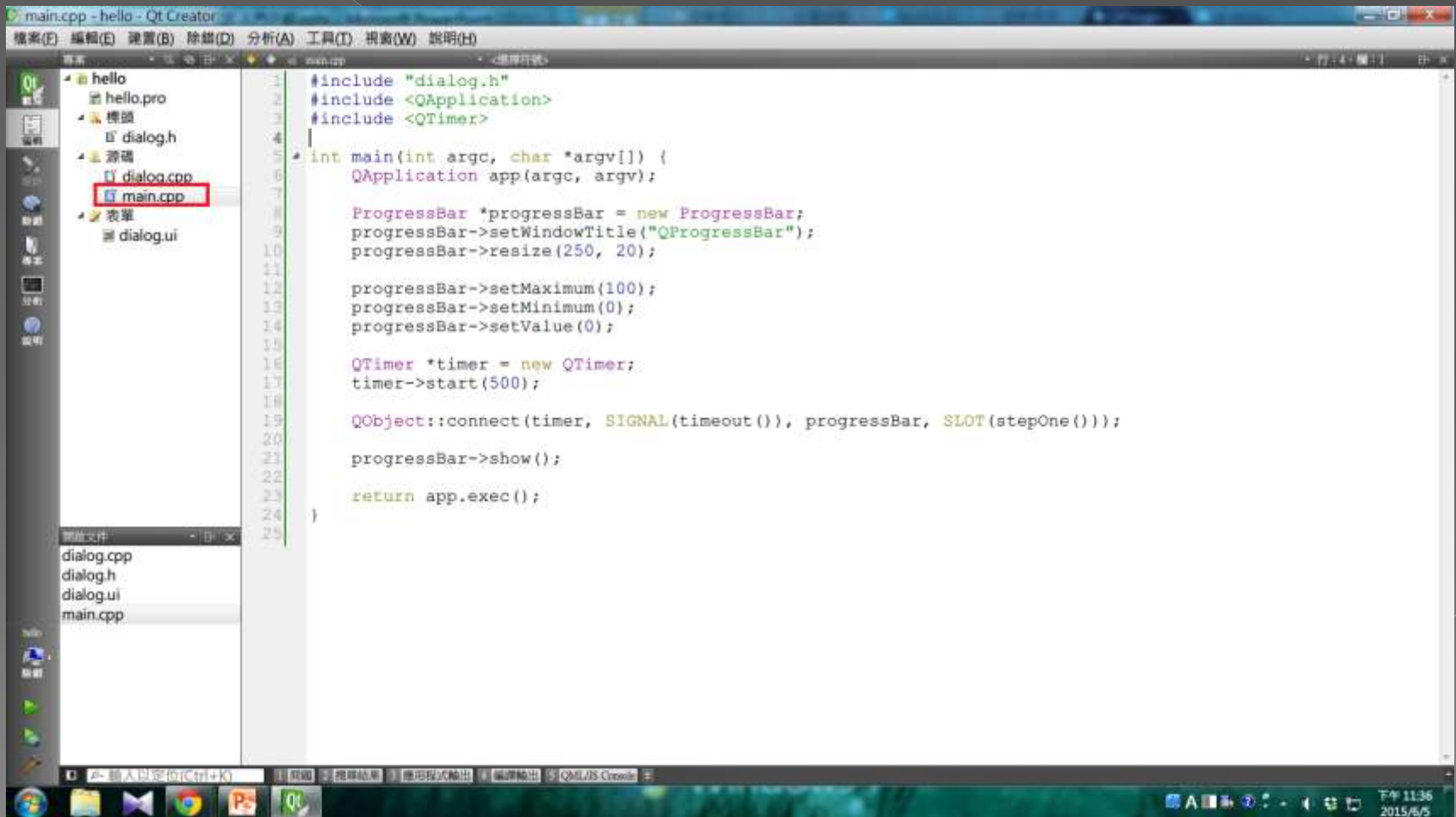
Progress Bar



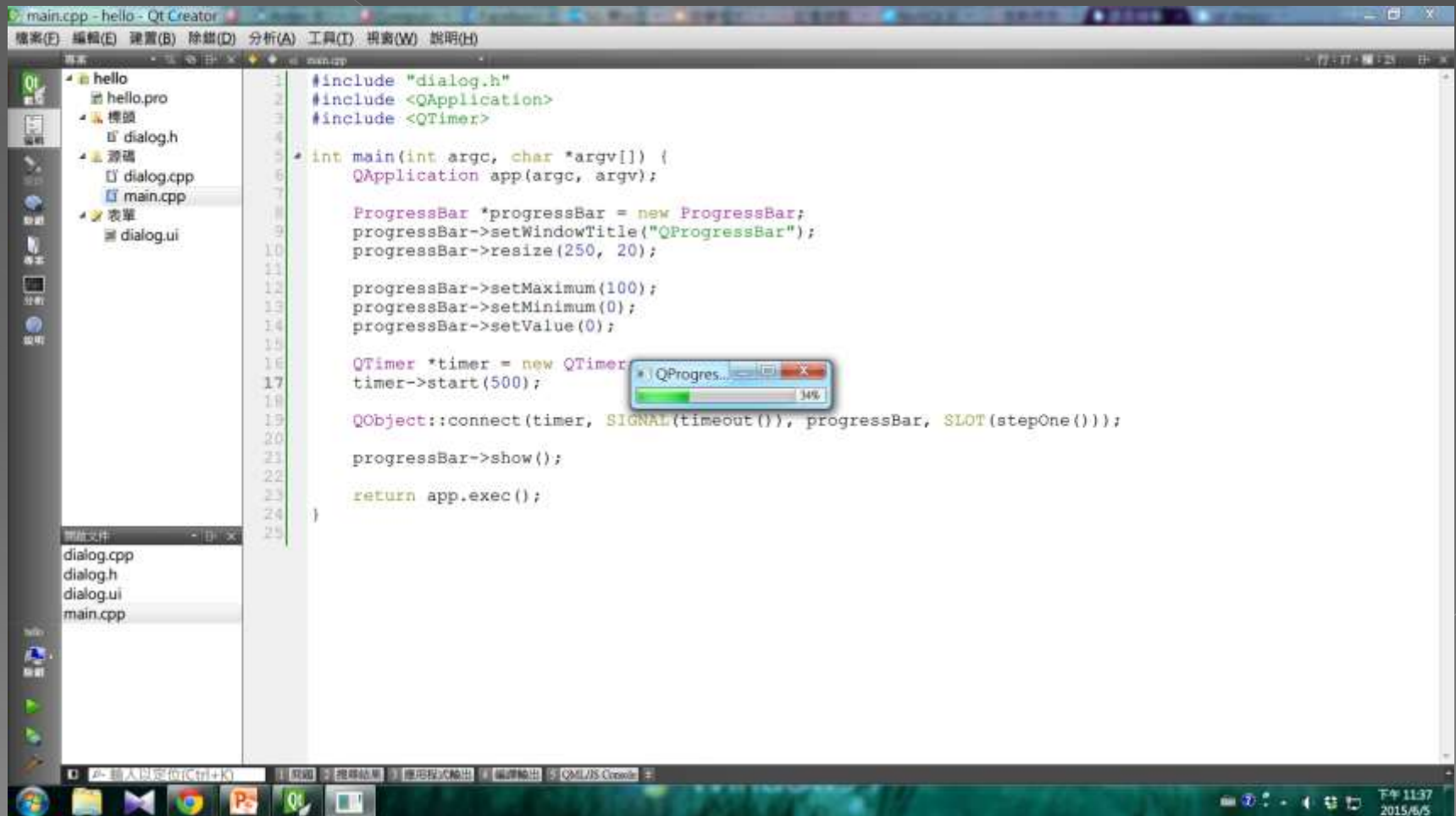
Progress Bar



Progress Bar



Progress Bar



Layout

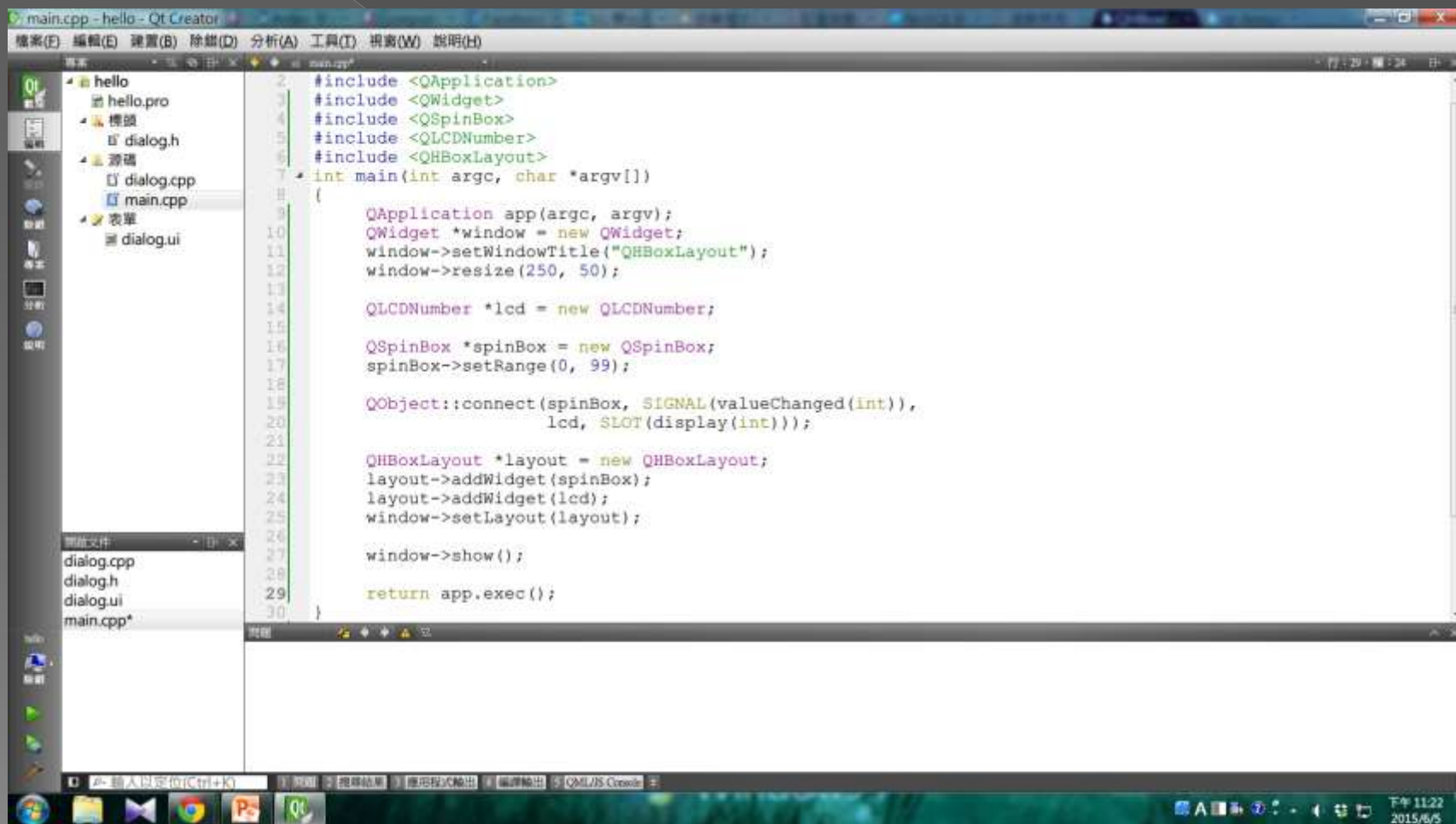


QHBoxLayout & QVBoxLayout

版面配置

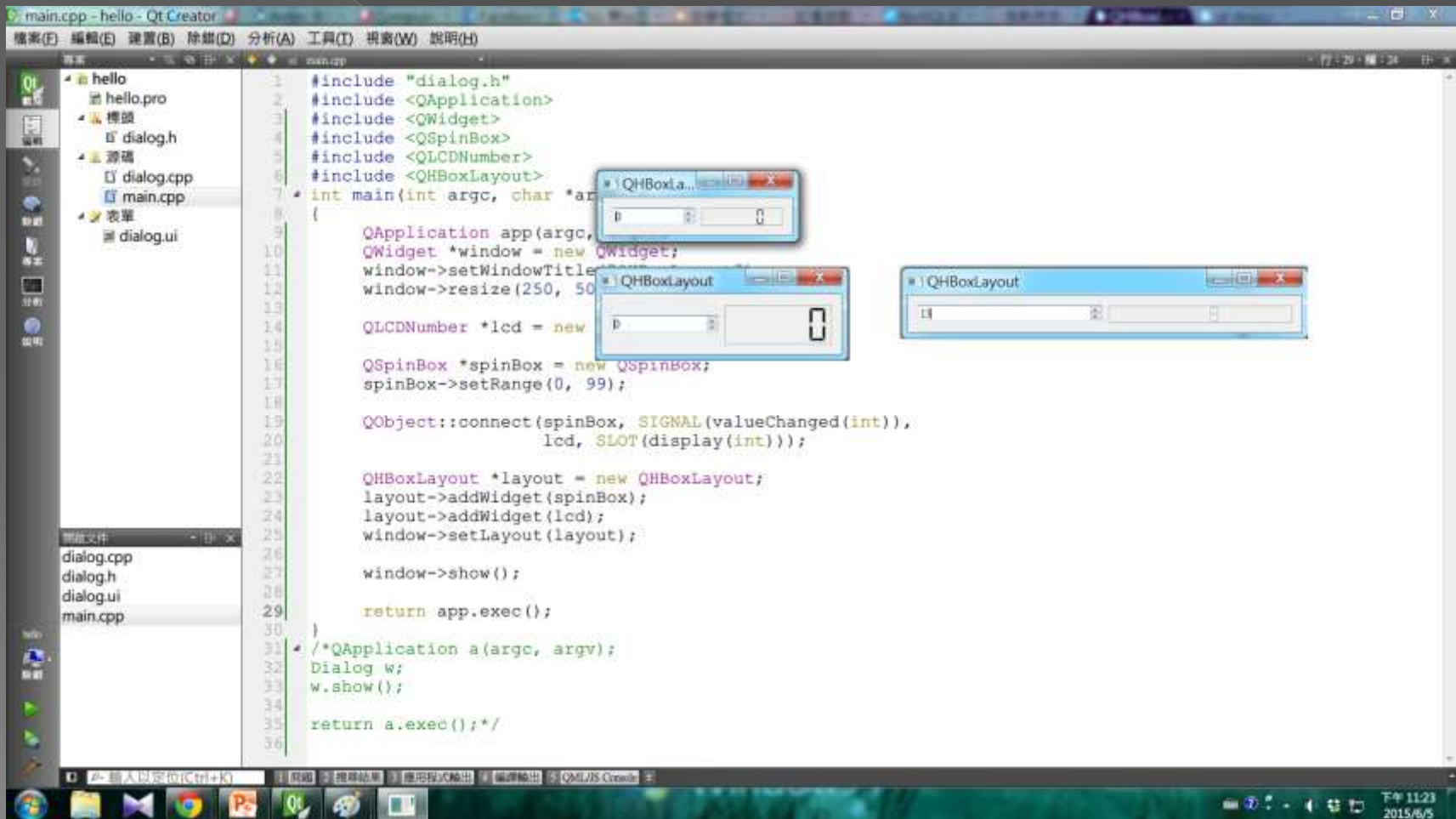
- 一般情況，在視窗縮放時，當中的元件位置並不會適當的自我調整大小、位置（或像是label字型大小自動調整之類的），以配合視窗縮放展現適當的觀感。
- 使用 QHBoxLayout 將元件加入，這會把 QHBoxLayout 及其管理的元件設成程式中 window 的子元件，並依 QHBoxLayout 版面配置策略自動水平配置元件
- QHBoxLayout 中元件的加入順序，就是水平配置由左至右顯示的順序

QHBoxLayout & QVBoxLayout 版面配置



QHBoxLayout & QVBoxLayout

版面配置



QML

QML

- ◉ Qt Modeling Language
- ◉ 基於JavaScript、宣告式編程的程式語言，用於設計使用者介面為主的應用程式
- ◉ QML 元素可以透過標準 JavaScript 增強，元素也可以無縫整合和使用 Qt 框架的 C++ 擴展。
- ◉ Cross platform
 - > Windows
 - > Linux
 - > Mac
 - > Android, IOS

QML 執行方式

- ◎ Linux

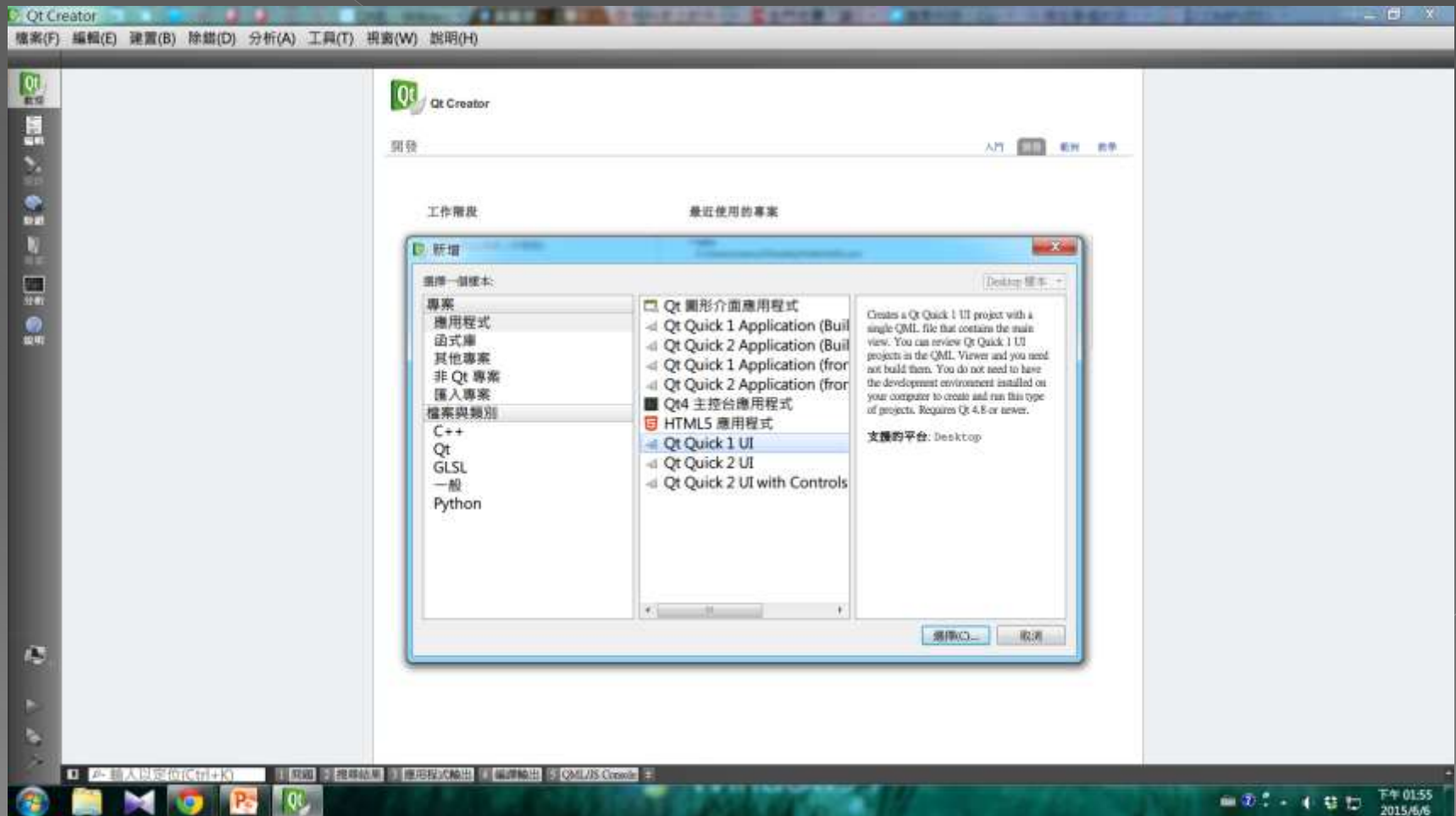
- > 直接裝 QT 執行

- ◎ Windows & Mac

- > If QML ver 1.X, QML viewer

- > If QML ver 2.X+, QML scene

Hello world



Hello world

第一個例子

```
import QtQuick 2.3
import QtQuick.Controls 1.3

ApplicationWindow {
    visible: true;
    width: 800;
    height: 600;
}
```

Hello world

第一個例子

一個 QML 檔案中
最基本的模組

```
import QtQuick 2.3
import QtQuick.Controls 1.3

ApplicationWindow {
    visible: true;
    width: 800;
    height: 600;
}
```

Hello world

第一個例子

```
import QtQuick 2.3
import QtQuick.Controls 1.3

ApplicationWindow {
    visible: true;
    width: 800;
    height: 600;
}
```

控制相關模組



Hello world

第一個例子

```
import QtQuick 2.3
```

```
import QtQuick.Controls 1.3
```

```
ApplicationWindow {  
    visible: true;  
    width: 800;  
    height: 600;  
}
```

元件類型 {
 屬性: 參數
 ...
}

Hello world

第一個例子

```
import QtQuick 2.3
```

```
import QtQuick.Controls 1.3
```

```
ApplicationWindow {
```

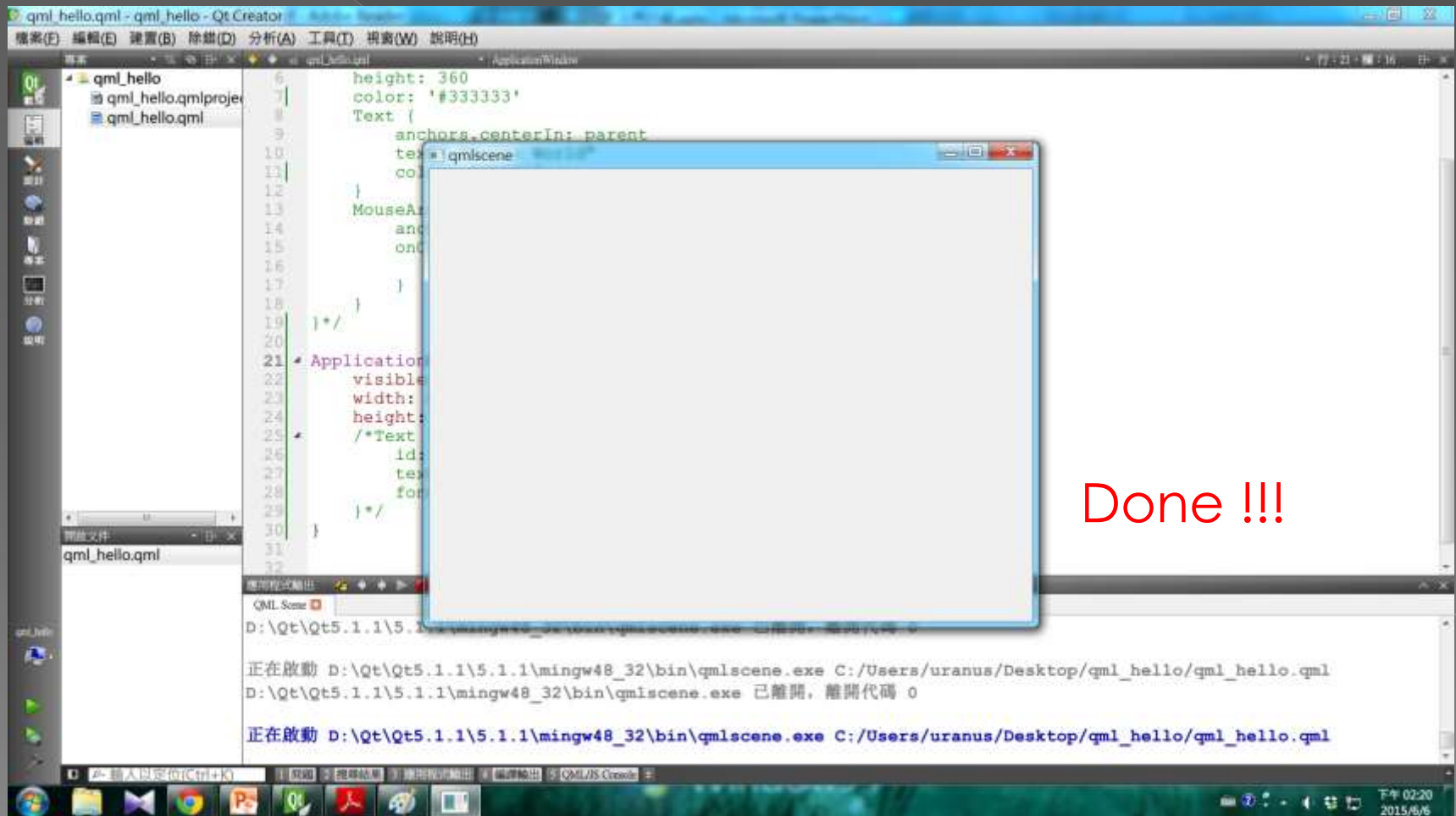
```
    visible: true;    → 視窗可顯示
```

```
    width: 800;        → 視窗寬度
```

```
    height: 600;       → 視窗高度
```

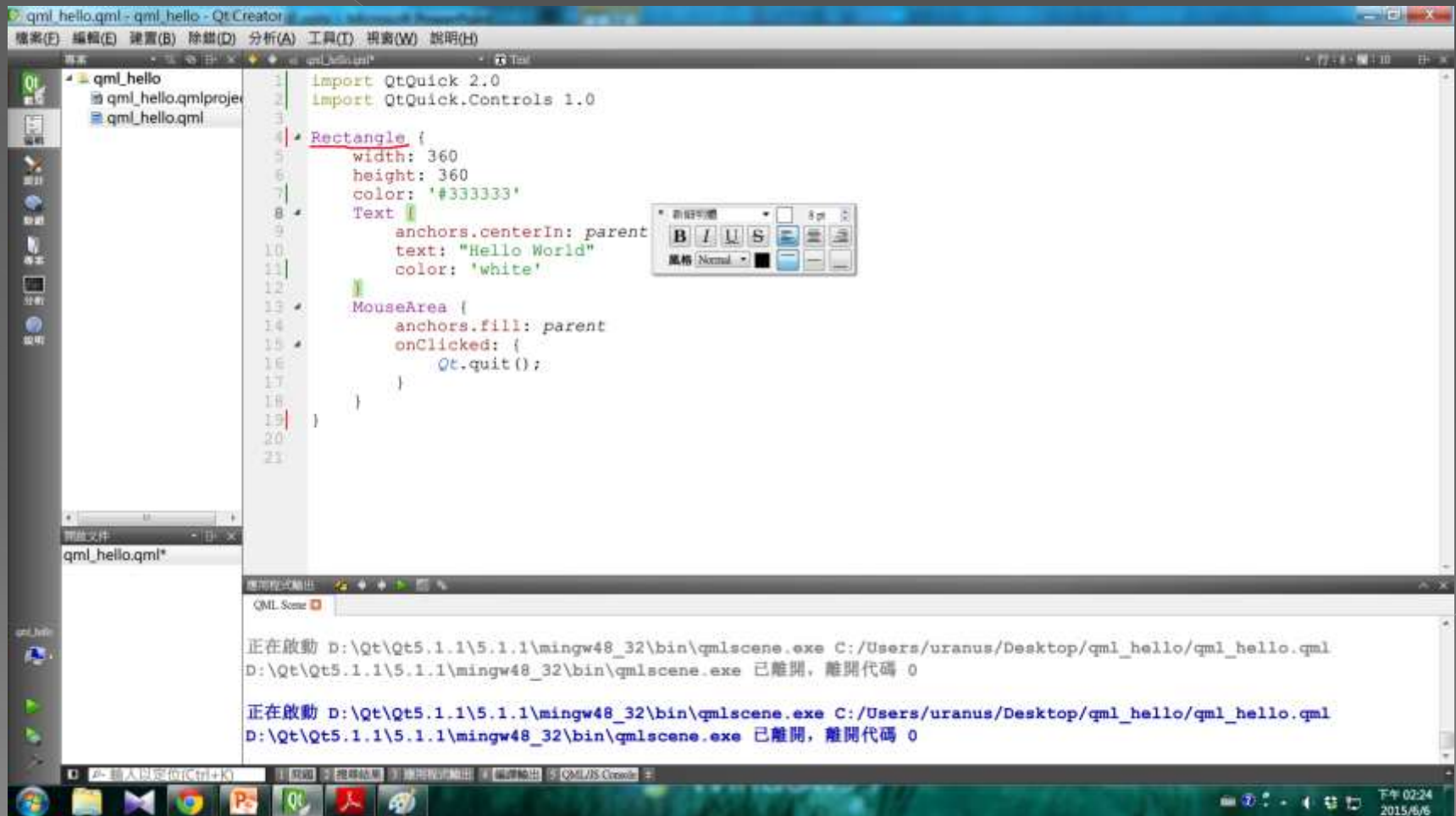
```
}
```


Hello world

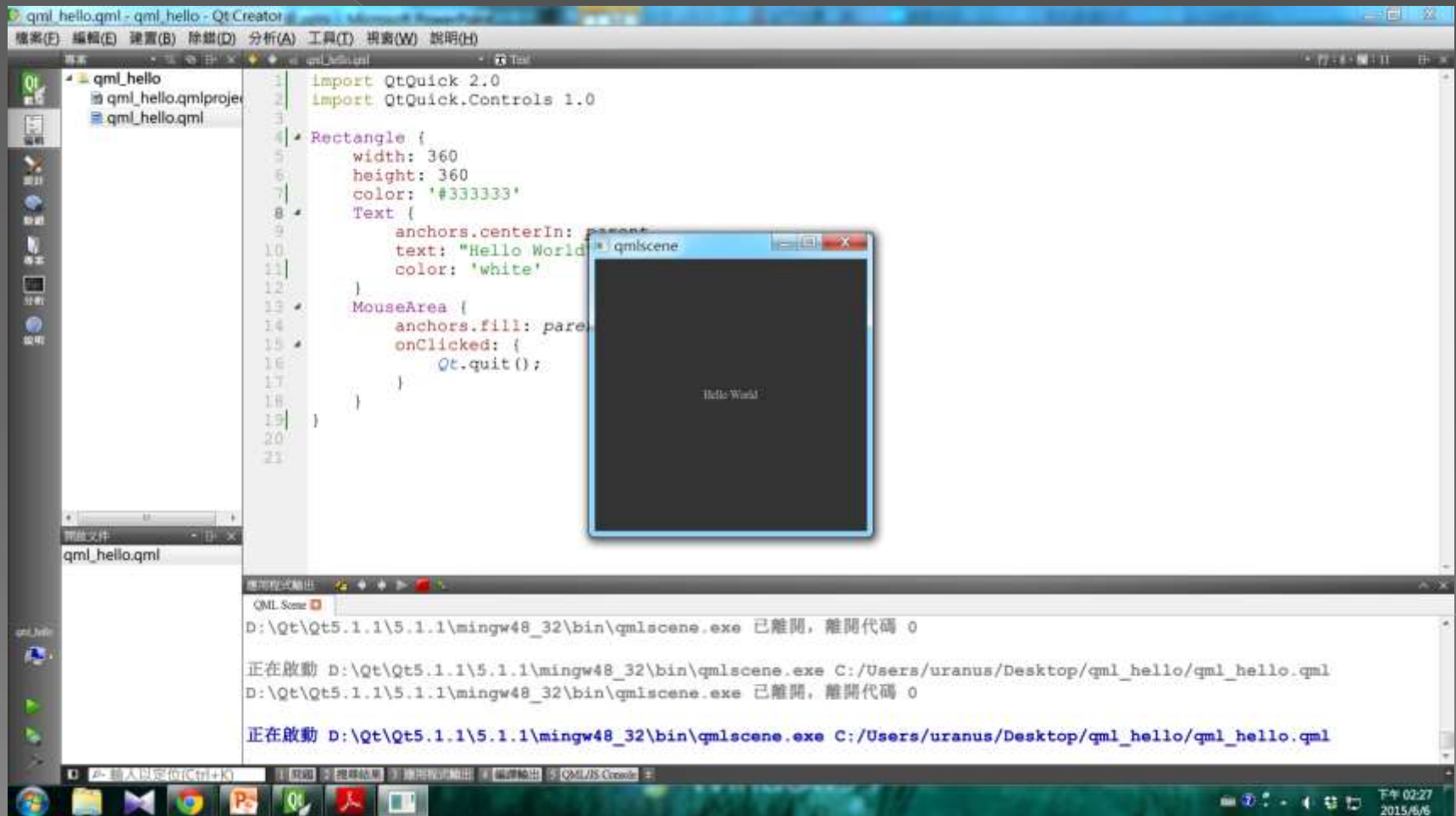


Done !!!

Hello world 2

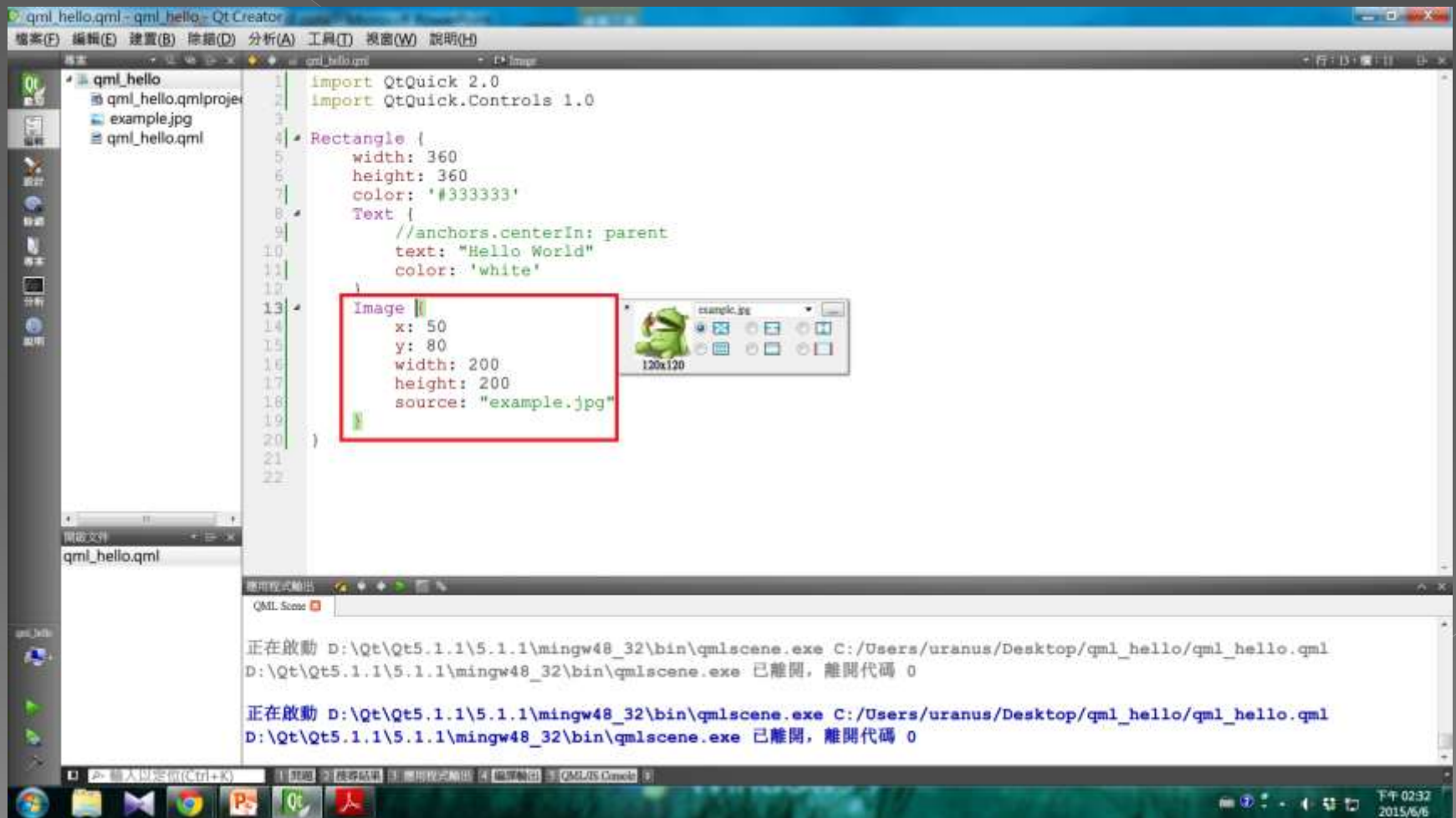


Hello world 2



Advanced QML

Create more !!



Create more !!



