



# QT Designer #1



## 핵심 정리

### 예제 Project

Project Name	DESIGNER_1
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp window.h



## 핵심 정리

- Window 클래스 안에는
  - ① UI(자식 윈도우들) 만들고 설정하는 코드와
  - ② 프로그램 자체의 기능을 구현하기 위한 코드가  
⇒ 섞여 있다.
- **UI관련 코드를 별도의 클래스로 분리.**
  - ⇒ 멤버 함수가 아닌 “**별도의 클래스로 분리**” 하면  
멤버 UI 관련 데이터도 분리할 수 있다.
  - ⇒ 클래스를 별도의 “**namespace Ui**” 로 분리
- **Ui::Form**
  - ⇒ UI를 만들기 위한 클래스
  - ⇒ Form 자체는 UI 가 아님.
  - ⇒ `setupUi()` 함수에서 `parent` 윈도우를 인자로  
받아서 해당 윈도우에 UI 구축.



## 핵심 정리

- `Ui::Form` 을 사용하는 2가지 방법
  - ① 다중 상속
  - ② 포함
- 이번 강의 핵심
  - ① UI 관련 코드는 `Ui::Form` 클래스로 분리
  - ② `Ui::Form` 클래스를 사용하는 2가지 방법
- UI 작업을 별도의 클래스로 분리해도, 직접 코드를 작성하는 것은 동일하다..
  - ⇒ 좀더 쉽게 할수 있을까 ?
  - ⇒ **.ui 파일과 designer 사용**



# QT Designer #2



## 핵심 정리

### 예제 Project

Project Name	DESIGNER_2
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp designer1.h designer2.h



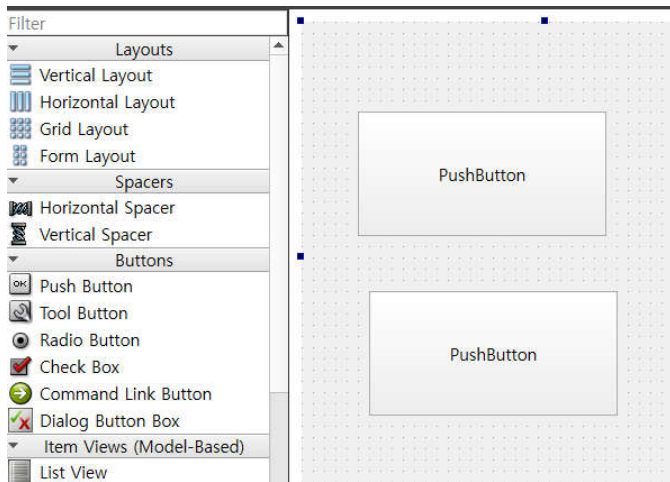
## 핵심 정리

- QT designer 를 사용해서 UI 만들려면
  - ⇒ 프로젝트에 designer 파일(.ui) 을 추가
  - ⇒ 강의에서는 “**form.ui**” 이름으로 추가



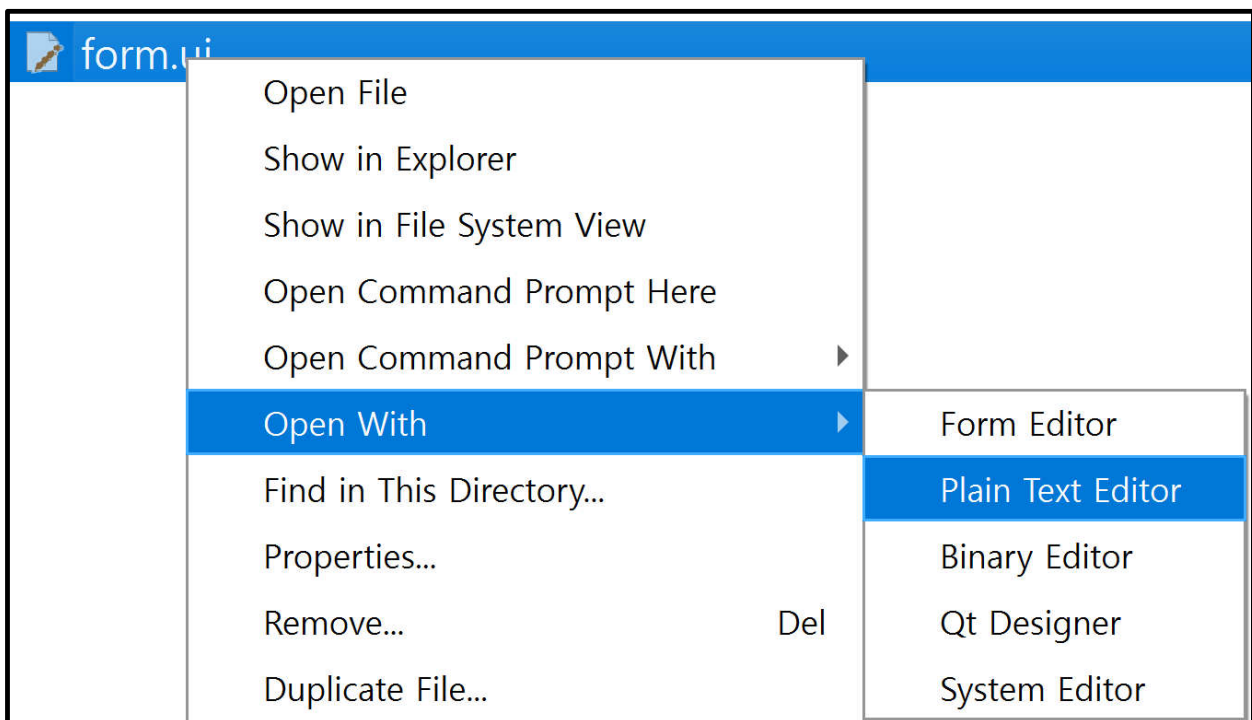
## 핵심 정리

- QT designer 파일(form.ui) 을 추가 하면
  - ⇒ .pro 에 “**FOMRS += form.ui**” 로 등록
  - ⇒ form.ui 파일 자체는 XML 형식의 파일



```
<widget class="QWidget" name="Form">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>390</width>
      <height>300</height>
    </rect>
  </property>
  <property name="windowTitle">
    <string>Form</string>
  </property>
  <widget class="QPushButton" name="pu:
    <property name="geometry">
      <rect>
        <x>80</x>
        <y>60</y>
        <width>171</width>
        <height>81</height>
```

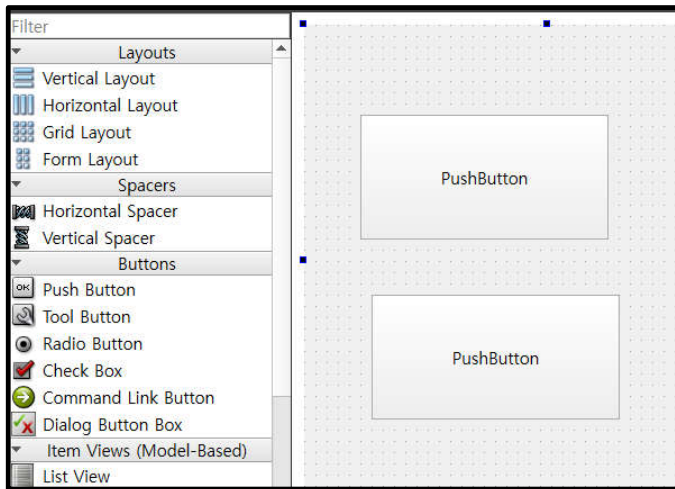
Form Editor 에서 편집하면 XML 이 변경







## 핵심 정리



```
<widget class="QWidget" name="Form">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>390</width>
      <height>300</height>
    </rect>
  </property>
  <property name="windowTitle">
    <string>Form</string>
  </property>
  <widget class="QPushButton" name="pushButton">
    <property name="geometry">
      <rect>
        <x>80</x>
        <y>60</y>
        <width>171</width>
        <height>81</height>
      </rect>
    </property>
  </widget>
</widget>
```

XML 자체는  
UI 디자인 시간에 생성

Meta Object Compiler

MOC 가  
XML 파일을 분석해서  
C++ 클래스를 생성한 것

```
namespace Ui
{
    class Form
    {
    }
}
```

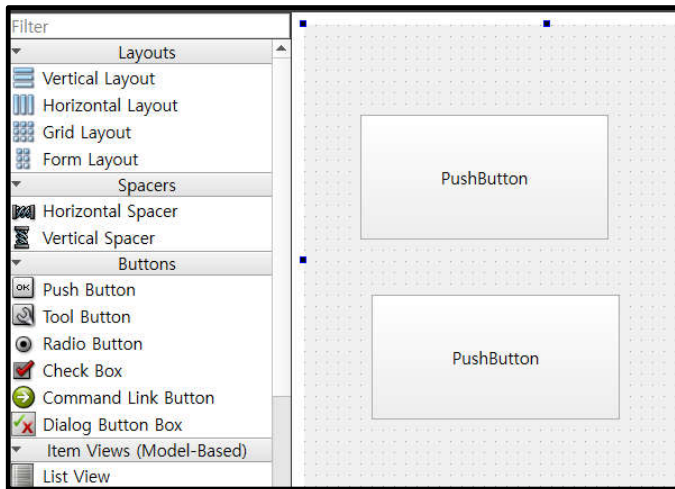
ui\_form.h



# Application(Qt) Project



# 핵심 정리



```
<widget class="QWidget" name="Form">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>390</width>
      <height>300</height>
    </rect>
  </property>
  <property name="windowTitle">
    <string>Form</string>
  </property>
  <widget class="QPushButton" name="pushButton">
    <property name="geometry">
      <rect>
        <x>80</x>
        <y>60</y>
        <width>171</width>
        <height>81</height>
      </rect>
    </property>
  </widget>
</widget>
```

**widget.ui**

**Meta Object Compiler**

```
namespace Ui
{
    class Widget
}

```

**ui\_widget.h**



## 핵심 정리

- **UI Design**

- ⇒ 컨트롤을 추가
- ⇒ 속성창 등을 사용해서 다양한 속성을 설정

- **Signal Slot 연결**

- ⇒ Form Editor 에서 연결
- ⇒ 코드에서 직접 연결