



# QT Widget Basic



## 핵심 정리

### 예제 Project

Project Name	WIDGET_1
Project Type	Application(Qt)
Base Class	QWidget

Vertical Layout

LCD Number **lcd**

Horizontal Slider **slider**

LineEdit **edit1**

LineEdit **edit2**

PushButton **button**

ComboBox **combo**



## 핵심 정리

- **핵심 1. 모든 Widgets 은 QWidget 으로 부터 파생된다.**
  - ⇒ QWidget 의 멤버 함수는  
모든 Widgets 의 공통의 특징이다.
  - ⇒ 모든 Widget은 setFont() 멤버 함수로 폰트를  
변경할 수 있다.
- **핵심 2. 각 Widgets 고유의 멤버 함수를 알아야 한다.**
  - ⇒ 속성을 변경하거나, 값을 얻거나/설정할 때 사용
  - ⇒ QT 공식 사이트의 다양한 예제 참고



- **핵심 3. QStyleSheet**

- ⇒ Widget의 다양한 속성을 “**css**” 스타일의 문법으로 설정하는 기술
- ⇒ 한 개 객체 뿐 아니라 동일 타입의 모든 객체에 적용도 가능

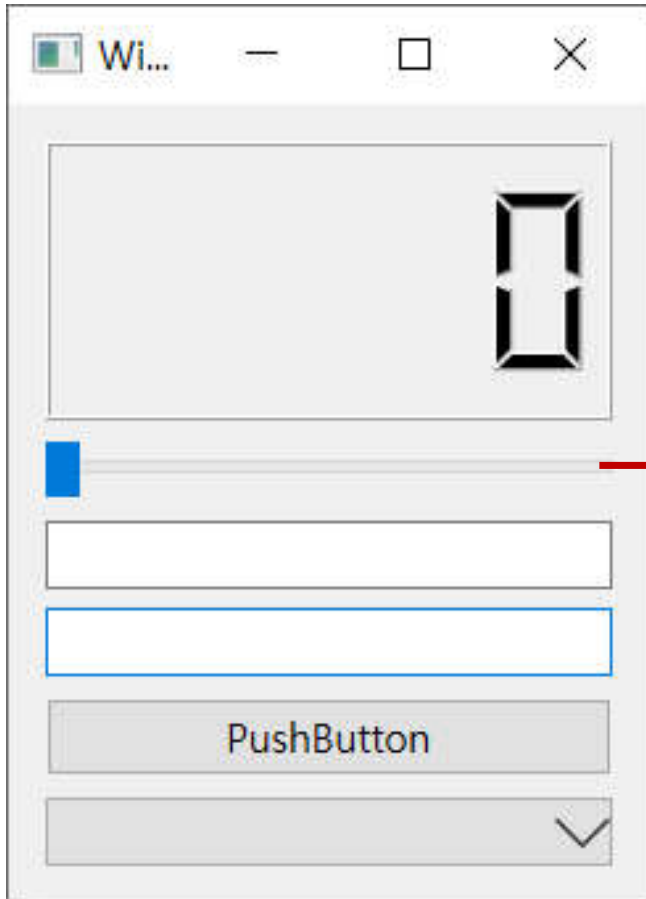
- **QT 공식 도움말에서**

- ⇒ “**Qt Style Sheets Examples**”
- ⇒ “**Qt Style Sheets Reference**”



## 핵심 정리

### ● 핵심 4. signal/slot



QLcdNumber 가 가지고 있는  
`display(int)` 라는  
slot 과 연결도 가능

슬라이더 값이 변경될 때 마다  
“`valueChaged(int)`” 라는  
signal 발생

사용자가 추가한  
slot 함수와 연결

```
void slider_chaged(int value)
{
}
```



QListWidget

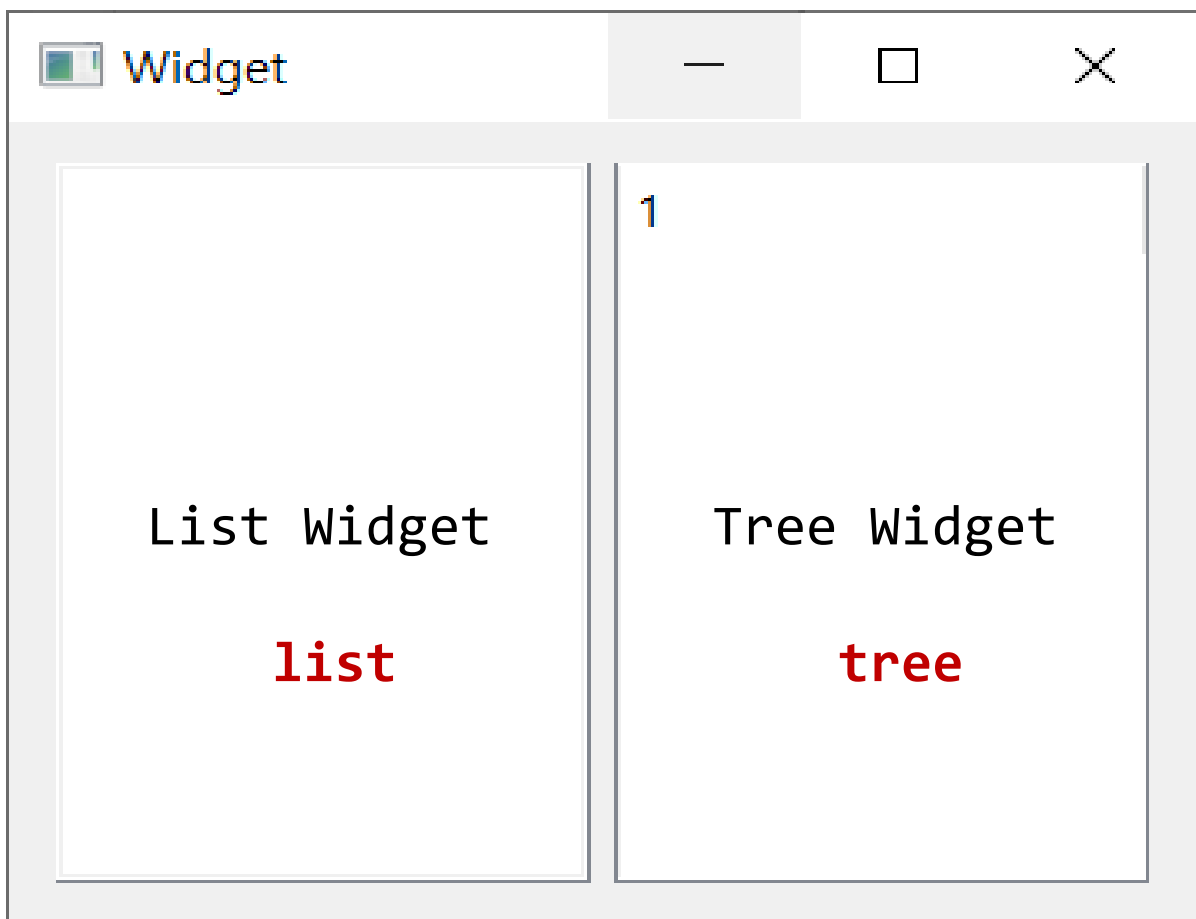
QTreeWidget



## 핵심 정리

### 예제 Project

Project Name	<code>LIST_TREE_WIDGET</code>
Project Type	<code>Application(Qt)</code>
Base Class	<code>QWidget</code>



**Layout Horizontal**



## 핵심 정리

- 핵심 1. 여러개의 값을 보관하는 Widget

⇒ ComboBox, List Widget, Tree Widget, Table Widget ...

⇒ “**항목의 타입**”을 정확히 알아 두어야 한다.

- QComboBox 에 항목을 추가하는 함수

```
addItem(const QString &text,  
        const QVariant &userData = QVariant());  
  
addItem(const QIcon &icon,  
        const QString &text,  
        const QVariant &userData = QVariant());
```

- QListWidget 에 항목을 추가하는 함수

```
addItem(const QString &label);  
addItem(QListWidgetItem *item);
```

항목의 타입이  
QListWidgetItem





# Subclassing



## 핵심 정리

### 예제 Project

Project Name

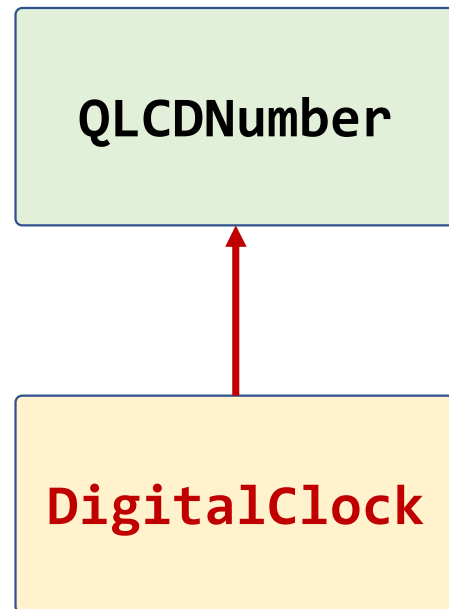
**SUBCLASSING**

Project Type

**Application(Qt)**

Base Class

**QWidget**



### ● Subclassing

- ⇒ QT 클래스로 부터 파생된 클래스를 만들어서 기능을 추가하는 개념.
- ⇒ “**QLCDNumber** 의 모든 기능 + 새로운 기능추가”
- ⇒ 객체지향 프로그램에서 널리 사용되는 개념.
- ⇒ 강의 예제는 “**QT 공식 사이트 예제**” 를 약간 변경한 것