**Основные виджеты и их свойства**

**Виджет QLabel**

Для внесение виджета **QLabel** в вашу форму, выполните следующее:



Сохраните дизайн в файл как **qlabel.ui** и конвертируйте его в файл **qlabel.py**. Теперь поработаем с ярлыком виджета при помощи кода.

***Изменение свойств QLabel:***

1. Чтобы **поменять шрифт QLabel**, используйте метод **setFont**().
2. Чтобы поменять размер **QLabel**, вам нужно указать его геометрию при помощи метода **setGeometry**() .
3. Чтобы изменить **текст в QLabel**, вы можете использовать метод **setText**().



**Виджет QLineEdit**

Виджет QLineEdit – это редактируемое поле, где вы можете принимать данные от пользователя. **LineEdit** содержит множество методов, с которыми можно работать.



Создаем новый дизайн **qline.ui**, используя **дизайнер PyQt5** и внести шесть виджетов **QLineEdit** и экспортируйте его в файл **qline.py**.

|  |  |
| --- | --- |
|  |  |

Методы **QLineEdit**:



Результат:



Метод **setStyleSheet**() может быть использован с виджетами PyQt5 для изменения стилей:

* Размер и тип шрифта;
* Цвет текста;
* Цвет заднего фона;
* Цвет верхней границы;
* Цвет нижней границы;
* Цвет левой границы;
* Цвет правой границы;
* Цвет выделения;
* Цвет выделения заднего фона.

**Виджет QPushButton**

О**бработка событий – это** взаимодействие виджета и функции.

**Событие клика в PyQt5** называется сигналом, и метод, который будет выполняться, называется **слот**. Так что при нажатии на **QPushButton**, сигнал издается. Названием сигнала в данном случае, является **clicked**(). Чтобы связать сигнал со слотом, нужно использовать метод **connect**(), что вы сейчас и увидите.

Создадим форму **myform.ui** при помощи [QLabel](https://python-scripts.com/pyqt5%22%20%5Cl%20%22qlabel-widget) и **QPushButton** и экспортируем её в файл **myform.py**.

|  |  |
| --- | --- |
|  |  |

**Визуальный редактор слота/сигнала**

Подключение сигнала виджета к слоту при помощи метода **connect**() - это не единственный способ.

На самом деле, для каждого виджета есть предопределенные слоты. Вы можете связать сигнал с любым предопределенным слотом, без необходимости кодить в дизайнере PyQt5.

Перетяните **QPushButton** и **QLineEdit** в вашу форму.

Нажмите F4 и перетяните курсор из QPushButton и отпустите его в верхней части QLineEdit - появится редактор сигнала/слота.



Чтобы вернуться в обычный режим, нажмите на F3.

Слева находятся предопределенные сигналы, справа — предопределенные слоты. Скажем, нам нужно подключить сигнал clicked() со слотом «*очистки содержимого*».

Выберите сигнал **clicked** слева и выберите **clear слот** справа и нажмите OK.



После выполнения подключений ваших сигналов и слотов, вы можете выйти из этого режима, нажав **ESC**, или **F3**.

Сейчас, если вы запустите эту форму, и нажмете **QPushButton**, то любой текст в **QLineEdit** будет очищен. Вы можете редактировать или удалять эти связи в панели редактора сигналов и слотов.

Сохраните форму как **signaledit.ui** и конвертируем её в **signaledit.py**:



**Виджет QComboBox**

Вместо того, чтобы разрешить пользователю вводить свои данные в [QLineEdit](https://python-scripts.com/pyqt5%22%20%5Cl%20%22QLineEdit-Widget), или любом другом редактируемом виджете, мы можем использовать **виджет QCombobBox**, чтобы дать список данных, из которого он сможет выбирать.

Перетянем **QComboBox** в нашу форму и взглянем на её методы.



Сохраняем файл как **combobox.ui** и конвертируем его в **combobox.py**:



По умолчанию **QComboBox** — пустой. Чтобы вносить объекты в **QComboBox**, используйте метод **addItem**():

**Получаем все элементы из QComboBox**

Нет прямого способа получить все значения из **QComboBox**, но вы можете применить [цикл Python](https://python-scripts.com/loops-for-while) для этой цели. Для этого подойдет функция [range](https://python-scripts.com/range%22%20%5Ct%20%22_blank).



|  |  |
| --- | --- |
|  | # по индексу, который начинается с нуля |

**Самостоятельно:**

Добавить в свою форму виджет по варианту и продемонстрировать максимальное количество свойств этого виджета:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вариант** | **1** | **2** | **3** | **4** | **5** |
| Компонент | Plain Text Edit | Text Edit | Text Browser | Progress Bar | Spin Box |
| **Вариант** | **6** | **7** | **8** | **9** | **10** |
| Компонент | GroupBox | RadioButton | Dock Widget | Stacked Widget | ToolBox |
| **Вариант** | **11** | **12** | **13** | **14** | **15** |
| Компонент | Graphics View | CheckBox | ListView | Tab Widget | Calendar Widget |
| **Вариант** | **16** | **17** | **18** | **19** | **20** |
| Компонент | Date Edit | Time Edit | Font Combo Box | Table View | Tree View |