**AMALIY MASHG‘ULOT UCHUN O‘QUV MATERIALLARI**

**2-Mavzu:** PyQt5 paketi va QtDesigner dasturi yordamida GUI dasturlarini yaratish**.**

**3-mashg‘ulot.** PyQt5 kutubxonasi. QLineEdit vidjeti.

**O‘quv savollari:**

1. QLineEdit vidjeti;
2. setStyleSheet() metodi.

**1. QLineEdit vidjeti**

**QLineEdit** - bu matnning bir qatorini kiritish va tahrirlash imkonini beruvchi vidjet. Ushbu vidjetda Cencel va Repeat, Cut va Past hamda “Sudrab olib tashlash” funksiyalari mavjud.

**import** **sys**

**from** **PyQt5.QtWidgets** **import** (QWidget, QLabel,

    QLineEdit, QApplication)

**class** **Examples**(QWidget):

**def** \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.initUI()

**def** initUI(self):

        self.lbl = QLabel(self)

        qlineE = QLineEdit(self)

        qlineE.move(100, 100)

        self.lbl.move(100, 40)

        qlineE.textChanged[str].connect(self.changed)

        self.setGeometry(300, 300, 280, 170)

        self.setWindowTitle(‘QlineEdit Example’)

        self.show()

**def** changed(self, text):

         self.lbl.setText(Label text)

        self.lbl.adjustSize()

**if** \_\_name\_\_ == ‘\_\_main\_\_’:

    app = QApplication(sys.argv)

    ex = Examples()

    sys.exit(app.exec\_())

Ushbu misolda QLineEdit vidjeti va QLabel ko‘rsatilgan. Tahrirlash paneliga kiritgan matnimiz darhol teg vidjetida ko‘rsatiladi.

qlineE = **QLineEdit**(**self**)

* QLineEdit vidjetining qline obyekti yaratildi.

qlineE.**textChanged**[str].**connect**(**self**.**changed**)

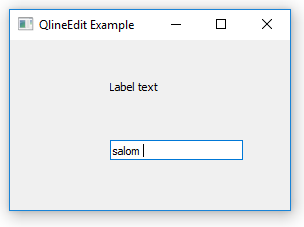
* Agar QLineEdit vidjetidagi matn o‘zgarsa, changed() funksiyasi ishga tushadi.

def **changed**(**self**, text):

**self**.lbl.**setText**(text Label)

**self**.lbl.**adjustSize**()

* changed funksiyasining ichida kiritilgan matnni teg vidjetiga o‘rnatish buyrug‘i joylashtirilgan. Yorliq o‘lchamini matn uzunligiga mos ravishda o‘zgartirish uchun adjustSize() metodidan foydalanilgan .



Yuqoridagi dastur ko‘rinishi

**PyQt QLineEdit**

PyQt QLineEdit bir qatorli matnli vidjet yaratish imkonini beradi. Odatda, siz QLineEditma'lumotlarni kiritish shaklida foydalanasiz .

Amalda siz ko'pincha QLineEditvidjetni vidjet bilan ishlatasiz QLabel.

Vidjet yaratish uchun QLineEditquyidagi amallarni bajaring.

Birinchidan, moduldan import QLineEditqiling PyQt6.QtWidgets:

from PyQt6.QtWidgets import QLineEditKod tili: Python ( python )

Ikkinchidan, QLineEditfoydalanadigan yangi ob'ekt yarating:

* Hech qanday dalil yo'q.
* Faqat ota-ona vidjeti bilan.
* Yoki birinchi argument sifatida standart satr qiymati bilan.

Masalan:

line\_edit = QLineEdit('Default Value', self)Kod tili: Python ( python )

Bundan tashqari, siz quyidagi qo'shimcha xususiyatlardan foydalanishingiz mumkin:

| **Metod** | **Turi** | **Ta’rifi** |
| --- | --- | --- |
| matn | ip | Satrni tahrirlash mazmuni |
| readOnly | Mantiqiy | To'g'ri yoki noto'g'ri. Agar rost boʻlsa, qatorni tahrir qilib boʻlmaydi |
| clearButtonEnabled | Mantiqiy | Aniq tugmani qo'shish to'g'ri |
| placeholderText | ip | Satrni tahrirlash bo'sh bo'lganda paydo bo'ladigan matn |
| maxLength | butun son | Maksimal kiritilishi mumkin bo'lgan belgilar sonini belgilang |
| echoMode | QLineEdit.EchoMode | Matnni ko'rsatish usulini o'zgartiring, masalan, parol |

PyQt QLineEdit vidjetiga misollar

Keling, vidjetdan foydalanishga misollar keltiraylik QLineEdit.

1) Oddiy PyQt QLineEdit misoli

Quyidagi dastur vidjetni qanday yaratishni ko'rsatadi QLineEdit:

import sys

from PyQt6.QtWidgets import (

QApplication,

QWidget,

QLineEdit,

QVBoxLayout

)

class MainWindow(QWidget):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

self.setWindowTitle('PyQt QLineEdit Widget')

self.setGeometry(100, 100, 320, 210)

search\_box = QLineEdit(

self,

placeholderText='Enter a keyword to search...',

clearButtonEnabled=True

)

*# place the widget on the window*

layout = QVBoxLayout()

layout.addWidget(search\_box)

self.setLayout(layout)

*# show the window*

self.show()

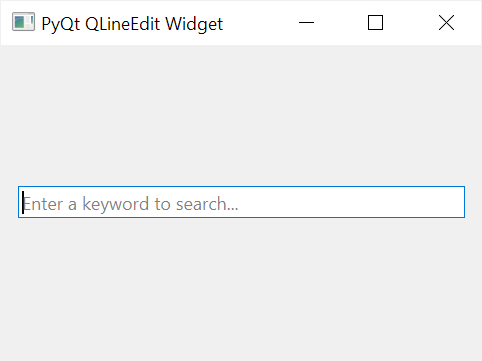
if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

window = MainWindow()

sys.exit(app.exec())Kod tili: Python ( python )

Chiqish:



2) Parol yozuvini yaratish uchun PyQt QLineEdit dan foydalanish

Quyidagi dastur QLineEditparol kiritish sifatida yangi vidjet yaratadi:

import sys

from PyQt6.QtWidgets import (

QApplication,

QWidget,

QLineEdit,

QVBoxLayout

)

class MainWindow(QWidget):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

self.setWindowTitle('PyQt QLineEdit Widget')

self.setGeometry(100, 100, 320, 210)

password = QLineEdit(self, echoMode=QLineEdit.EchoMode.Password)

*# place the widget on the window*

layout = QVBoxLayout()

layout.addWidget(password)

self.setLayout(layout)

*# show the window*

self.show()

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

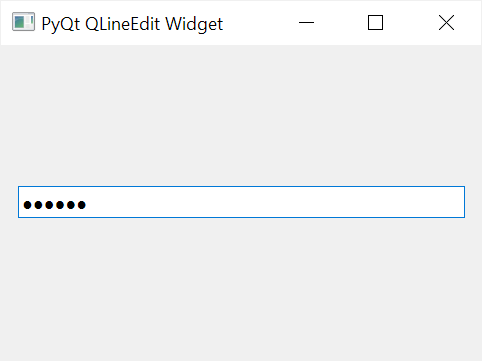
window = MainWindow()

sys.exit(app.exec())Kod tili: Python ( python )

QLineEditVidjetni parol kiritishiga aylantirish uchun siz echoModeni o'rnatasiz QLineEdit. EchoMode.Password:

password = QLineEdit(self, echoMode=QLineEdit.EchoMode.Password)Kod tili: Python ( python )

Chiqish:



3) PyQt QLineEdit-dan avtomatik to'ldirish xususiyati bilan foydalanish

Avtomatik to'ldirish xususiyati bilan yozuv yaratish uchun siz quyidagi amallarni bajaring:

Birinchidan, modulni import QCompleterqiling PyQt6.QtWidgets.

Ikkinchidan, QCompleteravtomatik toʻldirish funksiyasi uchun ishlatiladigan satrlar roʻyxati bilan vidjet yarating:

completer = QCompleter(word\_list)Kod tili: Python ( python )

Uchinchidan, a yarating va to'ldiruvchi ob'ekt bilan QLineEdituning usulini chaqiring :setCompleter()

line\_edit = QLineEdit(self)

line\_edit.setCompleter(completer)Kod tili: Python ( python )

Masalan, quyidagi dastur QLineEditavtomatik to'ldirish xususiyatiga ega vidjetni ko'rsatadi:

import sys

from PyQt6.QtWidgets import (

QApplication,

QWidget,

QLineEdit,

QVBoxLayout,

QCompleter

)

class MainWindow(QWidget):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

self.setWindowTitle('PyQt QLineEdit Widget')

self.setGeometry(100, 100, 320, 210)

common\_fruits = QCompleter([

'Apple',

'Apricot',

'Banana',

'Carambola',

'Olive',

'Oranges',

'Papaya',

'Peach',

'Pineapple',

'Pomegranate',

'Rambutan',

'Ramphal',

'Raspberries',

'Rose apple',

'Starfruit',

'Strawberries',

'Water apple',

])

fruit = QLineEdit(self)

fruit.setCompleter(common\_fruits)

*# place the widget on the window*

layout = QVBoxLayout()

layout.addWidget(fruit)

self.setLayout(layout)

*# show the window*

self.show()

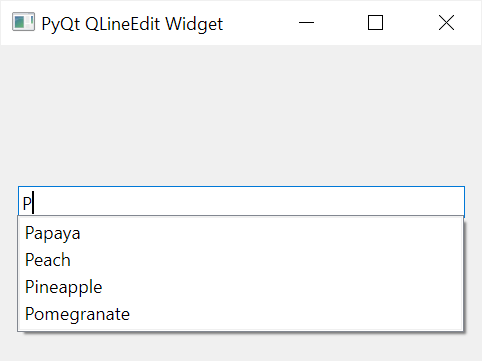
if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

window = MainWindow()

sys.exit(app.exec())Kod tili: Python ( python )

Chiqish:



Xulosa

* QLineEditBir qatorli kirish vidjetini yaratish uchun foydalaning .
* echoModeMatnni ko'rsatish usulini o'zgartirish uchun xususiyatdan foydalaning .
* QLineEditAvtomatik to‘ldirish funksiyasini qo‘llab-quvvatlash uchun vidjetdan QCompleter vidjeti bilan foydalaning .

**Nazorat savollari:**

1. PyQt da yorliq yaratish uchun qaysi vidgetdan foydalaniladi?
2. QLineEdit vidjetining vazifasi qanday?
3. QLineEdit vidjetining xususiyatlari qanday?