

First Part

در این درسنامه سه ویجت پاییهی PyQt6 را با جزئیات بیشتر بررسی می‌کنیم:

- **QLabel** (نمایش متن/تصویر)
- **QPushButton** (دکمه‌ی تعاملی)
- **QLineEdit** (فیلد تک‌خطی ورودی متن)

QLabel

خب QLabel ویجتی است برای نمایش متن یا تصویر با امکانات تنظیم تراز، تو رفتگی (indent) و تعامل دلخواه

[Qt Documentation](#)

ویژگی‌ها و متدهای مهم

- `text()` / `setText(str)`
- `setAlignment(Qt.Align...)` برای تنظیم تراز افقی/عمودی (Left, Right, Center, Justify) [GeeksforGeeks](#)
- `alignment()` برای واکنشی حالت تراز فعلی [GeeksforGeeks](#)
- `setIndent(int)` برای مشخص کردن تعداد پیکسل‌های تو رفتگی [Qt Documentation](#)
- `setPixmap(QPixmap)` برای نمایش تصویر به جای متن [Qt Documentation](#)
- `setTextInteractionFlags()` برای فعال کردن انتخاب متن یا لینک‌ها.

مثال کاربردی

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QLabel
from PyQt6.QtCore import Qt
from PyQt6.QtGui import QPixmap

app = QApplication(sys.argv)
```

```

8  window = QWidget()
9  window.setWindowTitle('QLabel Detailed Example')
10 window.resize(400, 200)
11
12 # create a text label
13 label_text = QLabel(window)
14 label_text.setText('Welcome to HDI OS') # set label text
15 label_text.setAlignment(Qt.AlignmentFlag.AlignCenter) #center-align text
16 label_text.setIndent(10) # 10px indent from
17 label_text.move(50, 20)
18 label_text.resize(300, 50)
19
20 # create an image label
21 label_image = QLabel(window)
22 pixmap = QPixmap('logo.png') # load image file
23 label_image.setPixmap(pixmap) # display image
24 label_image.setAlignment(Qt.AlignRight | Qt.AlignTop) # align image top-r
25 label_image.move(250, 80)
26 label_image.resize(100, 100)
27
28 window.show()
   sys.exit(app.exec())

```

QPushButton

خب QPushButton برای دریافت کلیک کاربر و اجرای عملیات مرتبط به آن طراحی شده است؛ از متن و آیکن تا کلید میانبر پشتیبانی می‌کند [Qt Documentation](#).

ویژگی‌ها و متدهای مهم

- `text()` / `setText(str)`
- `icon()` / `setIcon(QIcon)` برای افزودن آیکن [GeeksforGeeks](#).
- درج & در متن برای تعریف کلید میانبر (مثلاً `&Submit` → `Alt+S`) [Qt Documentation](#).
- سیگنال‌های:
 - `clicked()` هنگام کلیک (ماوس، اسپیس یا میانبر) [TutorialsPoint](#)

◦ `released()` , `pressed()` برای تشخیص فشردن و رهاشدن دکمه.

• `setShortcut(QKeySequence)` برای تعریف میانبر دلخواه.

مثال کاربردی

```

1  import sys
2  from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QMessageBox
3  from PyQt6.QtGui import QIcon
4
5  app = QApplication(sys.argv)
6  window = QWidget()
7  window.setWindowTitle('QPushButton Detailed Example')
8  window.resize(300, 200)
9
10 btn = QPushButton('&Download', window)      # Alt+D shortcut :contentRef
11 btn.setIcon(QIcon('download.png'))           # set button icon :contentF
12 btn.setToolTip('Click to download file')     # tooltip on hover
13 btn.move(80, 80)
14
15 def on_click():
16     QMessageBox.information(window, 'Info', 'Download started') # show n
17
18 btn.clicked.connect(on_click)                # connect clicked signal :cor
19
20 window.show()
21 sys.exit(app.exec())

```

QLineEdit

خب `QLineEdit` یک تک خط ورودی متن است که امکانات `placeholder`، محدودیت طول، ماسک ورودی و اعتبارسنجی را ارائه می‌کند [Qt Documentation](#) [Qt Documentation](#)

ویژگی‌ها و متدهای مهم

- `setPlaceholderText(str)` / `placeholderText()` [TutorialsPoint](#).

- `setMaxLength(int) / maxLength()` برای حداکثر طول متن [Qt Documentation](#).
- `setEchoMode(QLineEdit.EchoMode)` برای حالت پسورد یا مخفی‌سازی متن (Normal, Password, NoEcho) [Qt Documentation](#).
- `setInputMask(str) / inputMask()` برای تعریف ماسک ورودی (IP مثلاً تاریخ یا) [Stack Overflow](#).
- `setValidator(QValidator)` برای اعتبارسنجی دینامیک ورودی [Qt Documentation](#).
- `text() / setText(str) / displayText()` [Qt Documentation](#).
- `setAlignment(Qt.Align...)` برای تراز متن درون فیلد.

مثال کاربردی

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QLabel, QLineEdit, QPl
from PyQt6.QtGui import QIntValidator
from PyQt6.QtCore import Qt

app = QApplication(sys.argv)
window = QWidget()
window.setWindowTitle('QLineEdit Detailed Example')
window.resize(350, 180)

lbl = QLabel('Age:', window)
lbl.move(20, 30)

line = QLineEdit(window)
line.setPlaceholderText('Enter your age')           # placeholder text :conten
line.setMaxLength(3)                                # max 3 digits :contentf
line.setValidator(QIntValidator(0,120))              # only allow 0-120 :cor
line.setEchoMode(QLineEdit.Normal)                  # normal echo :contentf
line.setAlignment(Qt.AlignRight)                    # right-align input :cc
line.move(100, 25)
line.resize(200, 30)

btn = QPushButton('Submit', window)
btn.move(100, 80)

def submit():
```

```
27     if line.text() == '':
28         lbl.setText('Age: (required)')           # show error label
29         lbl.setStyleSheet('color: red')
30     else:
31         print('Entered age:', line.text())
32
33 btn.clicked.connect(submit)
34
35 window.show()
36 sys.exit(app.exec())
```

تمرین‌های تکمیلی

۱. حال QLabel را طوری تنظیم کن که روی کلیک آن سیگنالی صادر شده و در کنسول پیغام چاپ شود (ماوس‌پرس event).

۲. در QPushButton، از shortcut به‌جای & برای تعریف میانبر استفاده کن.

۳. برای QLineEdit ماسک تاریخ ("0000-00-00") تعریف کن و مطمئن شو که ورودی معتبر است.