

# Graphical User Interface

May 23, 2022

## 0.1 Graphical user interface (GUI) Qrafik İstifadəçi interfeysi

**PyQt5 - dərsləri** Təsnifatına görə interfeyslər : Qrafik istifadəçi interfeysi və Komanda sətiri interfeysi (COMMAND-LINE INTERFACE) olmaqla iki yerə bölünür.

İnterfeys - iki elementin qarşılıqlı təsirini təmin edən qovuşma nöqtəsi. Hesablama texnikasında çeşidli interfeyslər ən müxtəlif səviyyələrdə gerçəkləşdirilib: proqramlarla səmərəli işləməyə imkan verən ən aşkar qrafik istifadəçi interfeyslərindən (GUI) tutmuş, qurğularla kompüterin komponentlərinin qarşılıqlı əlaqəsini təmin edən, çox zaman görünməz olan, ancaq çox gərəkli aparat vasitələri interfeyslərinədək. İstifadəçi interfeysləri istifadəçinin kompüterlə səmərəli “ünsiyyətini” təmin edən qrafik təsvir sistemlərindən, komandalardan, sorğulardan və s. ibarət olur.

Biz python dili ilə qrafik istifadəçi interfeysi necə hazırlanmasına nəzər yetirəcəyik.İnterfeysi isə PyQt5 kitabxanası ilə hazırlayacağıq.Qt firmasının python2 və python3 üçün Pyqt5 kitabxanası açıq qaynaq kodlu olub GNU GPL V3 lisenziyasına sahibdir.

[ ]:

PyQt5 python dili ilə bərabər sistemə yüklənmədiyindən dolayı yükləməməizə ehtiyac var.

[2]: `# Unix/Linux`

```
sudo apt-get install python3-pyqt5
```

```
sudo apt-get install python3-pyqt5.qtsql
```

```
sudo apt-get install qttools5-dev-tools
```

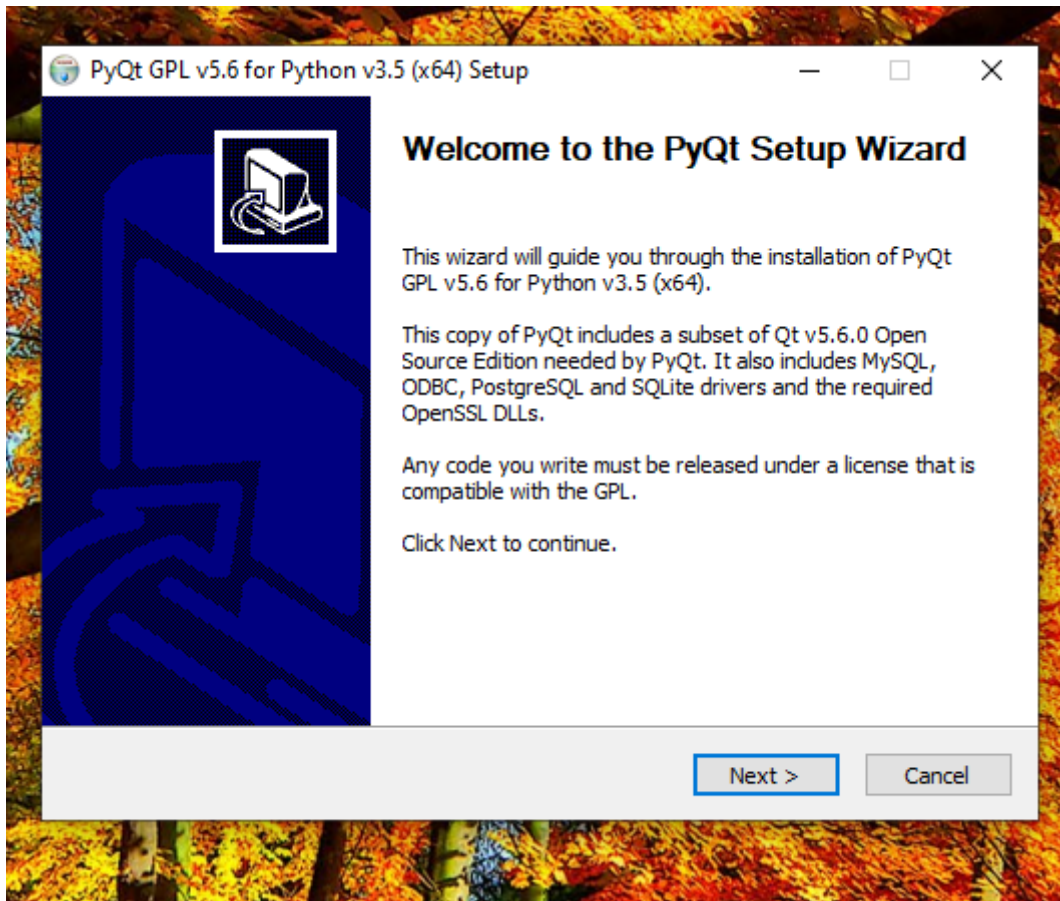
[6]: `#Windows`

Windows əməliyyat sistemi üçün aşağıdakı ünvandan kitabxananı yükləyə bilərsiniz.Qeyd edimki sisteminizə python dilini də yükləyəcəkdir. gər sistemdə python dili varsa aşağıdakı yükləməni həyata keçirməyin.Növbəti yükləmə metoduna nəzər yetirin

<https://sourceforge.net/projects/pyqt/files/PyQt5/PyQt5.6/PyQt5-5.6-gpl-Py3.5-Qt5.6.0-x64-2.exe/download>

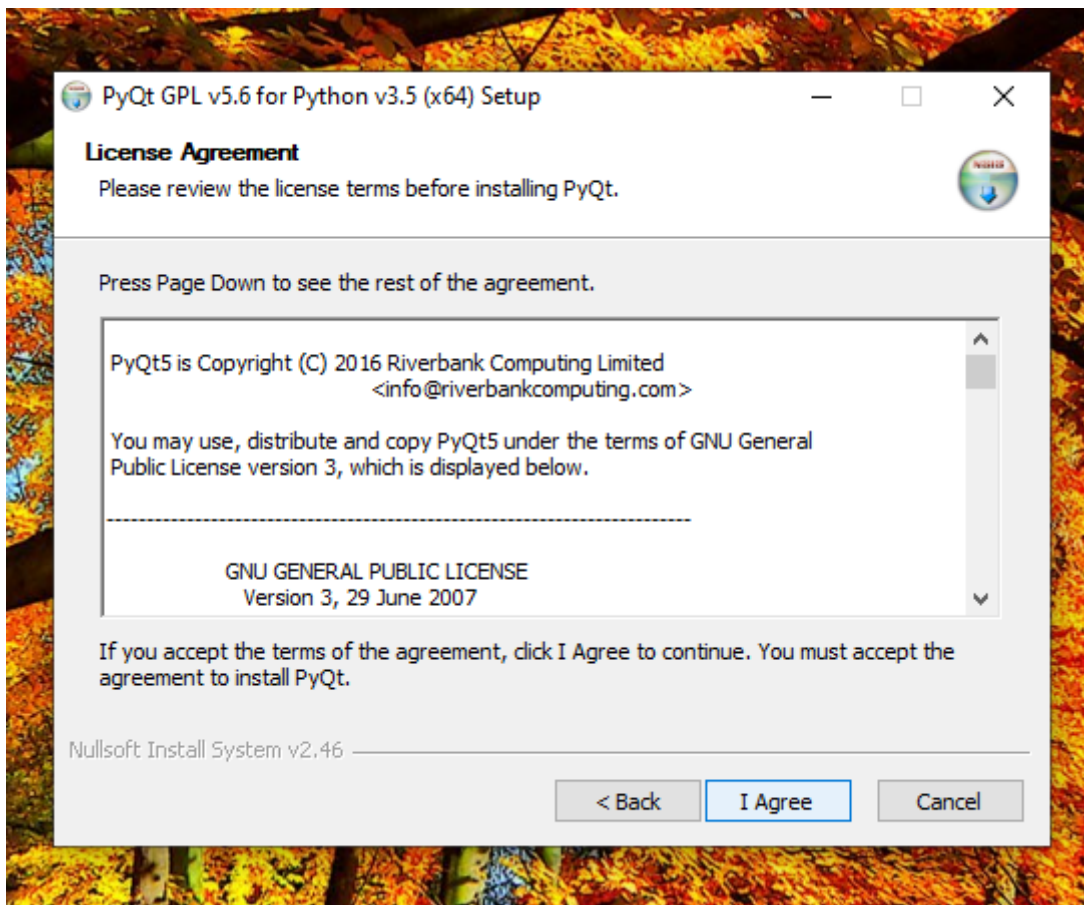
[7]: `from IPython.display import Image  
Image("../image/pyqt1.png")`

[7]:



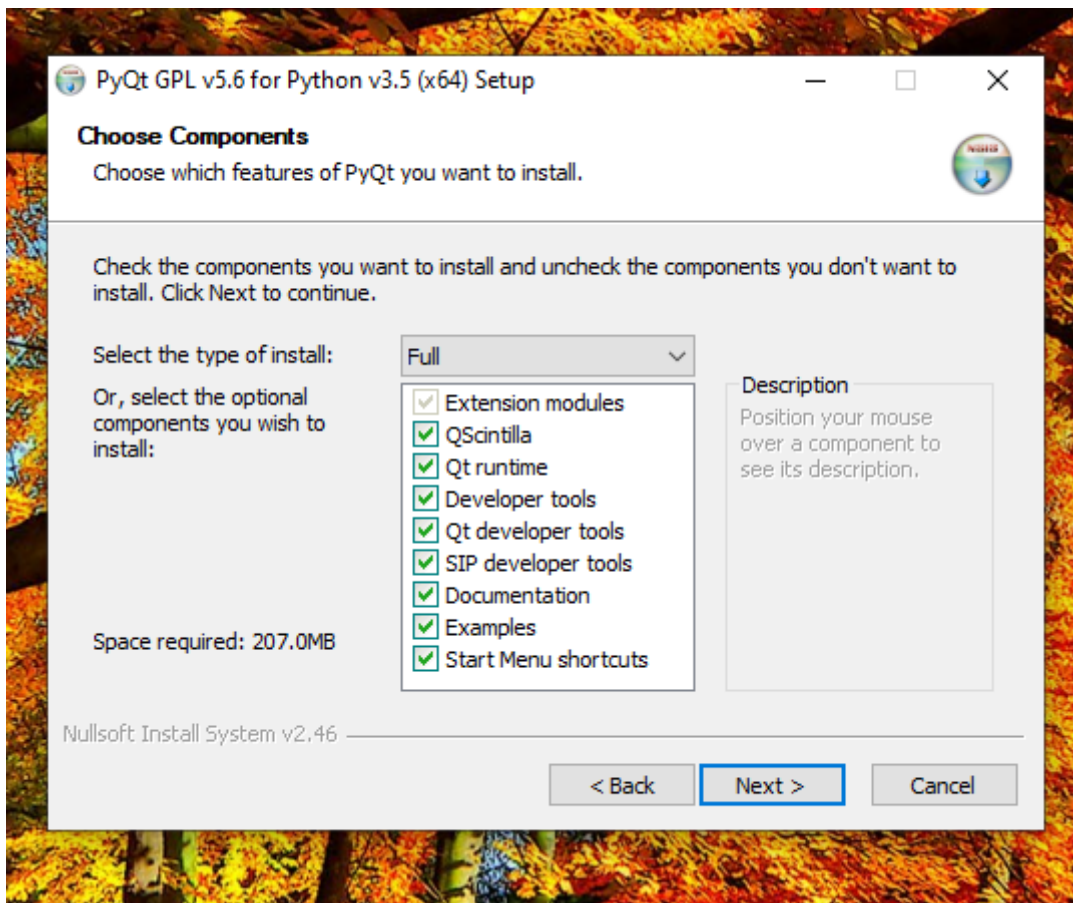
```
[8]: from IPython.display import Image
Image("../image/pyqt2.png")
```

[8]:



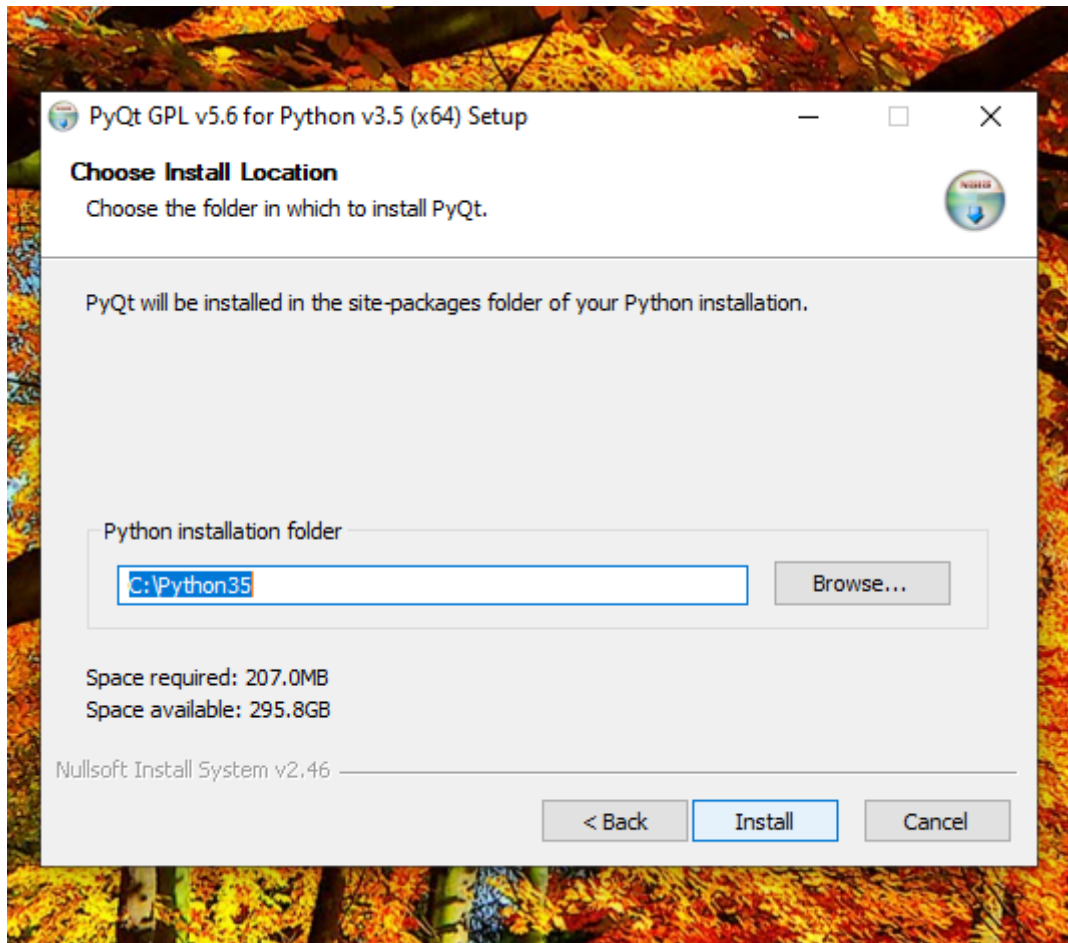
```
[9]: from IPython.display import Image  
Image("../image/pyqt3.png")
```

[9]:



```
[10]: from IPython.display import Image  
Image("../image/pyqt4.png")
```

[10]:



Və install deyərək həm kitabxananı həm də python 3.5 versiyasını sistemə yükləyə bilərsiniz.Yox əgər sisteminizdə anaconda platforması yüklüdürsə siz o zaman aşağıdakı yükləmə metodunu yerinə yetirin

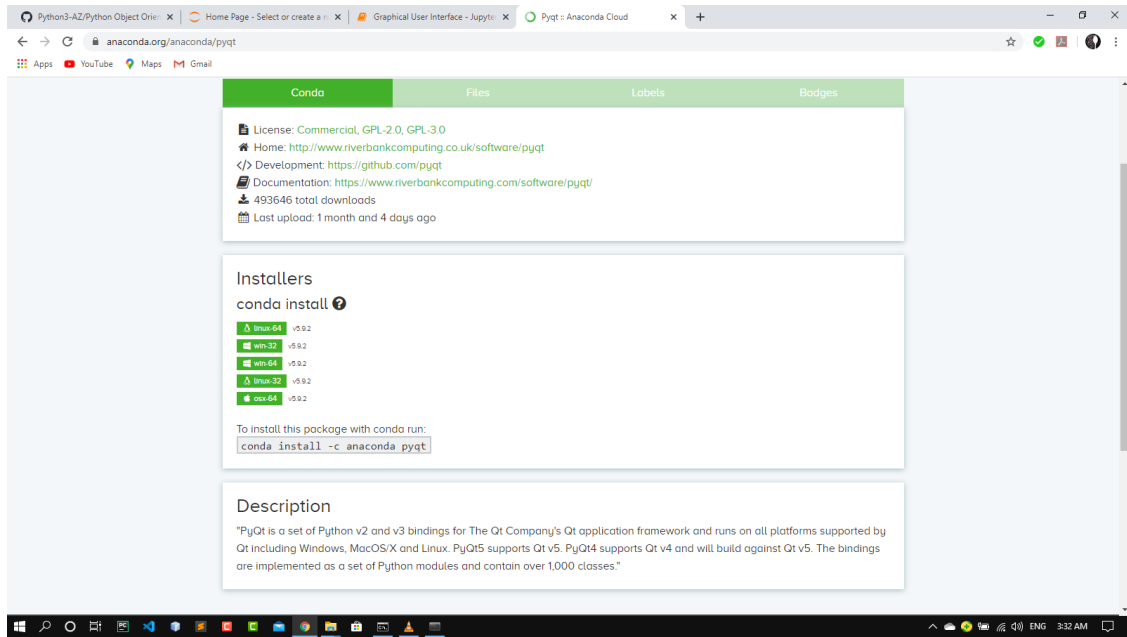
```
[11]: #Anaconda Platforms
```

```
[ ]:
```

```
conda install -c anaconda pyqt
```

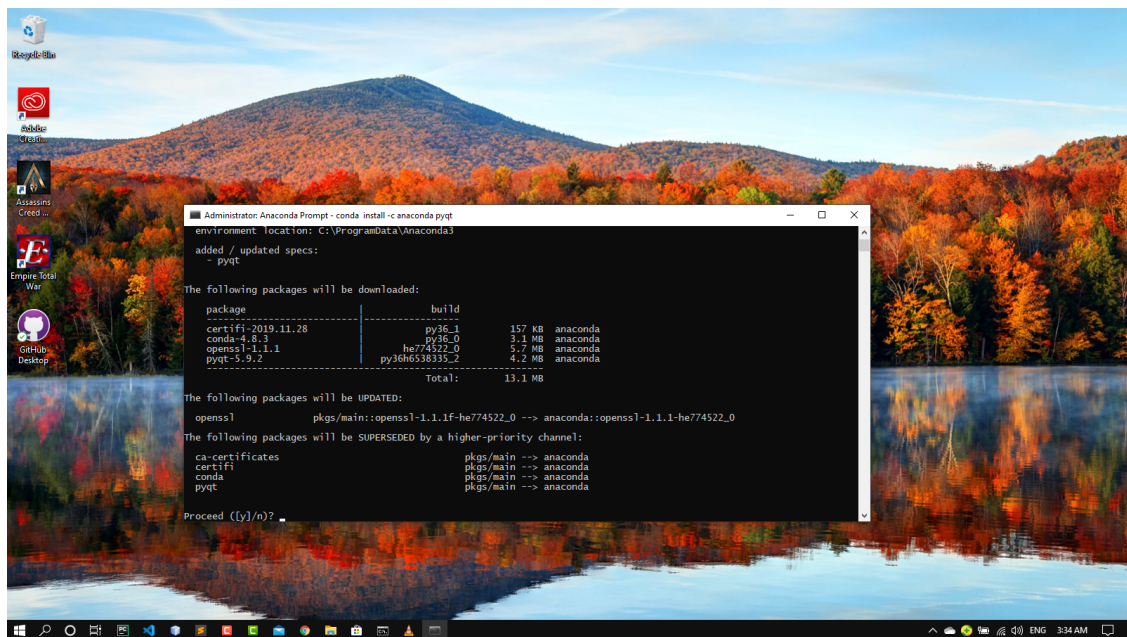
```
[12]: from IPython.display import Image  
Image("../image/pyanp.png")
```

```
[12]:
```



```
[13]: from IPython.display import Image
      Image("../image/qt.png")
```

[13]:



```
[16]: from IPython.display import Image
      Image("../image/qt1.png")
```

[16]:



```
base) C:\Windows\system32>pip install pyqt5
collecting pyqt5
Downloading PyQt5-5.14.1-5.14.1-cp35.cp36.cp37.cp38-none-win_amd64.whl (53.1 MB)
```

[ ]:

[17]: *# pip metodu ilə yükləmə*

pip install pyqt5

pip install pyqt5-tools

[19]: `from PyQt5 import QtCore`

import etdikdə xətası alt sətərə keçid etdik.PyQt daxilində siniflərdən təşkil olunmuş hissələrə ayrılır.Paketlər öz növbəsində müxtəlif icraçı rolunu oynayır

[21]: `print(dir(QtCore))`

```
['PYQT_CONFIGURATION', 'PYQT_VERSION', 'PYQT_VERSION_STR', 'QAbstractAnimation',
'QAbstractEventDispatcher', 'QAbstractItemModel', 'QAbstractListModel',
'QAbstractNativeEventFilter', 'QAbstractProxyModel', 'QAbstractState',
'QAbstractTableModel', 'QAbstractTransition', 'QAnimationGroup', 'QBasicTimer',
'QBitArray', 'QBuffer', 'QByteArray', 'QByteArrayMatcher', 'QCborError',
'QCborKnownTags', 'QCborSimpleType', 'QCborStreamReader', 'QCborStreamWriter',
'QChildEvent', 'QCollator', 'QCollatorSortKey', 'QCommandLineOption',
'QCommandLineParser', 'QConcatenateTablesProxyModel', 'QCoreApplication',
'QCryptographicHash', 'QDataStream', 'QDate', 'QDateTime', 'QDeadlineTimer',
'QDir', 'QDirIterator', 'QDynamicPropertyChangeEvent', 'QEasingCurve',
'QElapsedTimer', 'QEvent', 'QEventLoop', 'QEventLoopLocker', 'QEventTransition',
'QFile', 'QFileDevice', 'QFileInfo', 'QFileSelector', 'QFileSystemWatcher',
'QFinalState', 'QGenericArgument', 'QGenericReturnArgument', 'QHistoryState',
'QIODevice', 'QIdentityProxyModel', 'QItemSelection', 'QItemSelectionModel',
'QItemSelectionRange', 'QJsonDocument', 'QJsonParseError', 'QJsonValue',
'QLibrary', 'QLibraryInfo', 'QLine', 'QLineF', 'QLocale', 'QLockFile',
'QLoggingCategory', 'QMargins', 'QMarginsF', 'QMessageAuthenticationCode',
'QMessageLogContext', 'QMessageLogger', 'QMetaClassInfo', 'QMetaEnum',
'QMetaMethod', 'QMetaObject', 'QMetaProperty', 'QMetaType', 'QMimeData',
'QMimeDatabase', 'QMimeType', 'QModelIndex', 'QMutex', 'QMutexLocker',
'QObject', 'QObjectCleanupHandler', 'QOperatingSystemVersion',
'QParallelAnimationGroup', 'QPauseAnimation', 'QPersistentModelIndex',
'QPluginLoader', 'QPoint', 'QPointF', 'QProcess', 'QProcessEnvironment',
'QPropertyAnimation', 'QRandomGenerator', 'QReadLocker', 'QReadWriteLock',
'QRect', 'QRectF', 'QRegExp', 'QRegularExpression', 'QRegularExpressionMatch',
'QRegularExpressionMatchIterator', 'QResource', 'QRunnable', 'QSaveFile',
'QSemaphore', 'QSemaphoreReleaser', 'QSequentialAnimationGroup', 'QSettings',
'QSharedMemory', 'QSignalBlocker', 'QSignalMapper', 'QSignalTransition',
```

```
'QSize', 'QSizeF', 'QSocketNotifier', 'QSortFilterProxyModel', 'QStandardPaths',
'QState', 'QStateMachine', 'QStorageInfo', 'QStringListModel', 'QSysInfo',
'QSystemSemaphore', 'QT_TRANSLATE_NOOP', 'QT_TR_NOOP', 'QT_TR_NOOP_UTF8',
'QT_VERSION', 'QT_VERSION_STR', 'QTemporaryDir', 'QTemporaryFile',
'QTextBoundaryFinder', 'QTextCodec', 'QTextDecoder', 'QTextEncoder',
'QTextStream', 'QTextStreamManipulator', 'QThread', 'QThreadPool', 'QTime',
'QTimeLine', 'QTimeZone', 'QTimer', 'QTimerEvent', 'QTranslator',
'QTransposeProxyModel', 'QUrl', 'QUrlQuery', 'QUuid', 'QVariant',
'QVariantAnimation', 'QVersionNumber', 'QWaitCondition', 'QWinEventNotifier',
'QWriteLocker', 'QXmlStreamAttribute', 'QXmlStreamAttributes',
'QXmlStreamEntityDeclaration', 'QXmlStreamEntityResolver',
'QXmlStreamNamespaceDeclaration', 'QXmlStreamNotationDeclaration',
'QXmlStreamReader', 'QXmlStreamWriter', 'Q_ARG', 'Q_CLASSINFO', 'Q_ENUM',
'Q_ENUMS', 'Q_FLAG', 'Q_FLAGS', 'Q_RETURN_ARG', 'Qt', 'QtCriticalMsg',
'QtDebugMsg', 'QtFatalMsg', 'QtInfoMsg', 'QtMsgType', 'QtSystemMsg',
'QtWarningMsg', '__doc__', '__file__', '__license__', '__loader__', '__name__',
'__package__', '__spec__', 'bin_', 'bom', 'center', 'dec', 'endl', 'fixed',
'flush', 'forcepoint', 'forcesign', 'hex_', 'left', 'lowercasebase',
'lowercasedigits', 'noforcepoint', 'noforcesign', 'noshowbase', 'oct_',
'pyqtBoundSignal', 'pyqtPickleProtocol', 'pyqtProperty', 'pyqtRemoveInputHook',
'pyqtRestoreInputHook', 'pyqtSetPickleProtocol', 'pyqtSignal', 'pyqtSlot',
'qAbs', 'qAddPostRoutine', 'qAddPreRoutine', 'qChecksum', 'qCompress',
'qCritical', 'qDebug', 'qEnvironmentVariable', 'qErrnoWarning', 'qFatal',
'qFloatDistance', 'qFormatLogMessage', 'qFuzzyCompare', 'qInf', 'qInfo',
'qInstallMessageHandler', 'qIsFinite', 'qIsInf', 'qIsNaN', 'qIsNull', 'qQNaN',
'qRegisterResourceData', 'qRemovePostRoutine', 'qRound', 'qRound64', 'qSNaN',
'qSetFieldWidth', 'qSetMessagePattern', 'qSetPadChar',
'qSetRealNumberPrecision', 'qSharedBuild', 'qUncompress',
'qUnregisterResourceData', 'qVersion', 'qWarning', 'qrand', 'qsrand', 'reset',
'right', 'scientific', 'showbase', 'uppercasebase', 'uppercasedigits', 'ws']
```

Yuxarıda gördüyünüz kimi metod və funksiyalardan təşkil olunmuşdur

İlk pəncərəmizi hazırlayaq

```
[4]: #Funksiya daxilində hazırlayaq
```

```
[5]: from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

def Pencere():
    app = QApplication(sys.argv)
    window = QWidget()
    window.show()
    sys.exit(app.exec_())
Pencere()
```



```
An exception has occurred, use %tb to see the full traceback.
```

```
SystemExit: 0
```

sas interfeys siniflərdən təşkil olunub. Daxilində bazaların idarə olunması rəsmlər, animasiya tipli qrafik göstərimlər və sairə

Üç əsas pəncərə : QWidget , QMainWindow və QDialog

QtCore - qrafik interfeyslə əlaqəsi olmayıb, arxa planda çalışdırılacaq əməli prosesləri təmin edir. Daxilində digər modulları əhatə edir. QtGui - klas qrafik interfeyslə birbaşa əlaqəlidir (user interface). Rəsm və videolar üzərində təyinatı əməlləri təşkil edir. QtSql - verilənlərin bazası ilə əlaqədar olub, bazanın yaradılması, əlavələri təşkil edir. Misal üçün bazalar: SQLite, MySQL, Pl/SQL və s. QtSvg - vektorların təyininə istifadə olunan moduldur. (SVG) QtOpenGL - OpenGL dəstəklənməsi QtNetwork - şəbəkə ilə əlaqəli sinifdir. QtMultimedia - multimedia üzərində dəyişiklərdə istifadə olunur. QtXml - xml faylların düzəlişində istifadə olunur. QtScript (QtScriptTools) - Skriptlərin istifadəsi üçün nəzərdə tutulub. Qt - bu sinif bütün modullara, parametrlərə qoşulur. Designer -> Qrafik interfeys Assistant -> Qt sənədləri arasında keçid Linguist -> tərcüməçi

QMainWindow: Status bar və menu bar'ı olan bir pəncərə QWidget: Bütün widgetləri dəstəkləyən təməl pəncərə hazırlayır QDialog: Dialog - istifadəçi ilə dialoq qurulacaq şəkildə pəncərə hazırlayır

```
[6]: #Pəncərənin adını dəyişək
```

```
[7]: from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

def Pencere():
    app = QApplication(sys.argv)
    window = QWidget()
    window.setWindowTitle('Mənim ilk proqramım')
    window.show()
    sys.exit(app.exec_())
Pencere()
```

```
An exception has occurred, use %tb to see the full traceback.
```

```
SystemExit: 0
```

```
C:\ProgramData\Anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3339: UserWarning: To exit: use
'exit', 'quit', or Ctrl-D.
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

```
[3]: #və ya
```

```
[4]: from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

def Pencere():
    app = QApplication(sys.argv)
    window = QWidget()
    window.setWindowTitle('Mənim ilk programım')
    window.show()
    sys.exit(app.exec_())
if __name__ == '__main__':
    Pencere()
```

An exception has occurred, use %tb to see the full traceback.

SystemExit: 0

C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3339: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

```
[8]: #Qeyd edimki grafik interfeys proqramları,skript kimi çalışdırmağınız
      ↪məsləhətlidir.
      # Funksiyalardan əlavə ,biz daimi istifadə edəcəyimiz obyekt yönümlü
      ↪proqramlaşdırmadan istifadə metodudur.
```

```
[9]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.

app = QApplication(sys.argv)
window = MainWindow()
```

```
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()
```

[9]: 0

Pəncərəni sağa-sola darta bilirik,böyüdürük kiçildirik.Narahat olmayın bütün bunların öhdəsindən gəlib fiks-sabit pəncərə hazırlayacağıq.Pəncərənin əsas adı python olaraq qeyd olunub,gəlin onu dəyişək

[10]: *#setWindowTitle*

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()
```

[1]: 0

[1]: *#Pəncərə daxilində yazı göstərək*  
*#Label*

```
[5]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        # sas pəncərə self ifadəsində referans alındı.
```

```

        self.setWindowTitle("Bu mənim ilk proqramım")
        self.label = QLabel('PyQt dərsləri',self) #self ifadəsi MainWindow
        ↪pəncərədə olduğunu təmsil edir
app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

[5]: 0

```

[2]: #Yazını mərkəzə əlavə etmək
      # setAlignment()
      # setCentralWidget(label)

```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        self.label = QLabel('PyQt dərsləri',self) #self ifadəsi MainWindow
        ↪pəncərədə olduğunu təmsil edir
        self.label.setAlignment(Qt.AlignCenter) #Yazını mərkəzləşdiririk
        self.setCentralWidget(self.label) #Pəncərənin mərkəzində göstərilməsi
        ↪təmin olunur

app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

```

[1]: #setText()

```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

```

```

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
        ↪təmsil edir
        self.label.setText('Pyqt Dərsləri')
        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)
app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

[ ]:

[2]: #Yazı rəngləri

```

[3]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
        ↪təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<font color='Blue'>Pyqt dərsləri</font>")
        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)
app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

[3]: 0

```
[4]: #Pəncərə arxa plan rəngi
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        #Pəncərə arxa plan rəngi dəyişdirildi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background,Qt.darkGray)
        self.setPalette(self.palette)
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
        ↪təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<font color='Blue'>PyQt dərsləri</font>")
        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)

app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()
```

```
[1]: 0
```

```
[2]: #setWindowIcon - Pəncərə logosu
```

```
[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
```



```

        #Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        #Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background,Qt.white)
        self.setPalette(self.palette)
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
↪təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<font color='Blue'>PyQt dərsləri</font>")
        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)

app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

```
[1]: #resize parametri
```

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        #Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        #Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background,Qt.white)
        self.setPalette(self.palette)
        #resize
        self.resize(512,512) #element resize(x,y) x-width y-height
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
↪təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<font color='Blue'>PyQt dərsləri</font>")

```

```

        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)

app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

[2]: 0

[5]: *#move koordinat oxları üzrə yazını yerləşdirmək*  
*#Tətbiq etmək üçün yazını mərkəzləmə üsulunu silməliyik*

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        #Pəncərə logosu
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı
        ↪ küncdə logo əlavə olundu
        #Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        #resize
        self.resize(512,512) #element resize(x,y) x-width y-height
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu
        ↪ təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<font color='Blue'>PyQt dərsləri</font>")
        self.label.move(150,50) # move(x,y) x, y -coordinate

app = QApplication(sys.argv)
window = MainWindow()
window.show() #Pəncərənin göstərilməsi
# Programın davamlı göstərilməsi
app.exec_()

```

```
[1]: # setGeometry(x,y,width,height) Pəncərənin İşçi stolunda verilən parametrlərə  
      ↪ uyğun görsətmək
```

```
[2]: # pəncərə en və uzunluğu da geometry metodunda yer aldığından resize metodunu  
      ↪ silməliyik
```

```
[3]: # kitabxanaları proqrama daxil edirik  
from PyQt5.QtWidgets import *  
from PyQt5.QtCore import *  
from PyQt5.QtGui import *  
import sys  
  
class MainWindow(QMainWindow):  
  
    def __init__(self, *args, **kwargs):  
        super(MainWindow, self).__init__(*args, **kwargs)  
        # sas pəncərə self ifadəsində referans alındı.  
        self.setWindowTitle("Bu mənim ilk proqramım")  
        #Pəncərə logosu  
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı  
        ↪kündə logo əlavə olundu  
        #Pəncərə arxa plan rəngi  
        self.palette = QPalette()  
        self.palette.setColor(QPalette.Background,Qt.white)  
        self.setPalette(self.palette)  
        #resize  
        #self.resize(512,512) #element resize(x,y) x-width y-height  
        #setGeometry(x,y,width,height)  
        self.setGeometry(600,150,512,512)  
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu  
        ↪təmsil edir  
        #self.label.setText('Pyqt Dərsləri')  
        self.label.setText("<font color='Blue'>PyQt dərsləri</font>")  
        self.label.move(150,50) # move(x,y) x, y -coordinate  
  
app = QApplication(sys.argv)  
window = MainWindow()  
window.show() #Pəncərənin göstərilməsi  
# Programın davamlı göstərilməsi  
app.exec_()
```

```
[3]: 0
```

```
[ ]:
```

Yuxarıdakı nümunələri QMainWindow pəncərəsi üzərində göstərdik.İndi gəlin QWidget pəncərəsi hazırlayıb əlavələrimizi edək

```
[9]: #setAlignment() metodu
#Metod,ifadəni sağ,sol,mərkəz və yana əymək parametrlərini alır.
↳ (AlignLeft,AlignRight,AlignCenter,AlignJustify,AlignBottom)
```

```
[5]: #Metoddan biraz öncə istifadə etmişdik
```

```
[10]: #QLabel linkactivated
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        #sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        #Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı↳
↳kündə loqo əlavə olundu
        #Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background,Qt.white)
        self.setPalette(self.palette)
        #resize
        #self.resize(512,512) #element resize(x,y) x-width y-height
        #setGeometry(x,y,width,height)
        self.setGeometry(600,150,512,512)
        self.label = QLabel(self) #self ifadəsi MainWindow pəncərədə olduğunu↳
↳təmsil edir
        #self.label.setText('Pyqt Dərsləri')
        self.label.setText("<A href='Github'>GOOGLE</A>")
        self.label.move(50,50) # move(x,y) x, y -coordinate
        self.label.linkActivated.connect(self.click)
        self.show()

    def click(event=None):
        webbrowser.open_new(r"https://github.com/RashadGarayev/Python3-AZ")

if __name__ == '__main__':
    app = QApplication([])
    gui =MainWindow()
    app.exec_()
```

```
[2]: #QLabel QPixmap
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.label=QLabel(self)
        self.pixmap = QPixmap('../image/ai.png')
        self.label.setPixmap(self.pixmap)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

```
[2]: #linkActivated metodu (QLabel)
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        #sas pəncərə self ifadəsində referans alındı.
```

```

        self.setWindowTitle("Bu mənim ilk proqramım")
        #Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) #Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        #Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background,Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.label = QLabel(self) #self ifadəsi QWidget pəncərədə olduğunu
        ➔təmsil edir
        self.label.setText("<A href='Github'>GOOGLE</A>")
        self.label.move(50,50) # move(x,y) x, y -coordinate
        self.label.linkActivated.connect(self.click)
        self.show()

    def click(event=None):
        webbrowser.open_new(r"https://github.com/RashadGarayev/Python3-AZ")

if __name__ == '__main__':
    app = QApplication([])
    gui =MainWindow()
    app.exec_()

```

[ ]:

[2]: #linkHovered

```

[3]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)

```



```

        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.label=QLabel(self)
        self.label.setText("<A href='www.google.com'>GOOGLE</A>")
        self.label.linkHovered.connect(self.goster)
        self.show()
    def goster(self):
        print(u'Məni göstər')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Məni göstər

```
[4]: #QLineEdit class (sinif)
```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```
[3]: #Sinfin ala bildiyi metodlar
      #Qt.AlignCenter
```

```
[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setAlignment(Qt.AlignCenter) #kursoru mərkəzdə göstərir
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

```
[1]: #setEchoMode() metodu
      #Metod daxilində dörd parametrlə alır.
      #1.QLineEdit.Normal
      #2.QLineEdit.NoEcho
      #3.QLineEdit.Password
      #4.QLineEdit.PasswordEchoOnEdit
```

```
[2]: #QLineEdit.Normal
```

```
[3]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser
```

```

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineEdit = QLineEdit(self)
        self.lineEdit.setAlignment(Qt.AlignLeft)#kursoru solda göstərir
        self.lineEdit.setEchoMode(QLineEdit.Normal)
        self.lineEdit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[4]: #QlineEdit.NoEcho parametri  
#Parametr qutunu fəaliyyətsiz vəziyyətə gətirir

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()

```

```

        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setAlignment(Qt.AlignLeft)#kursoru solda göstərir
        self.lineedit.setEchoMode(QLineEdit.NoEcho)#Parametr qutunu
        ↪fəaliyyətsiz vəziyyətə gətirir.
        self.lineedit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: *#QlineEdit.Password parametri*

[2]: *#Parametrdən şifrələrin soruşulması və kənardan görünməsinin qarşısının*  
 ↪*alınması üçün istifadə olunur.Qutuya daxil olan ifadəni ulduz işarələri*  
 ↪*altında gizlədir.*

[3]: *# kitabxanaları proqrama daxil edirik*

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setAlignment(Qt.AlignLeft)#kursoru solda göstərir
        self.lineedit.setEchoMode(QLineEdit.Password)#Password.
        self.lineedit.move(200,50)
        self.show()

```

```

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[4]: *#QLineEdit.PasswordEchoOnEdit parametri*  
*#Parametrdən şifrə qutusunun açıq şəkildə, ifadənin görünməsi üçün istifadə*  
*→ olunur. Bundan əvvəlki parametr ifadənin qapalı (ulduz işarələri) olmasını*  
*→ tənzimləyirdisə, bu parametr isə tamamilə açıq şəkildə tənzimləyir.*

[5]: *#Bu metoda tes-z-tez rast gəlirik. Şifrəni daxil edirik görünməsini təmin etmək*  
*→ üçün aşağıda checkbox -u qeyd edirik.*

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        →kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineEdit = QLineEdit(self)
        self.lineEdit.setAlignment(Qt.AlignLeft) #kursoru solda göstərir
        self.lineEdit.setEchoMode(QLineEdit.PasswordEchoOnEdit) #Password.
        self.lineEdit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```
[1]: #setMaxLength() metodu
#Parametr qutuya daxil olan maksimum ifadə sayını tənzimləyir.Parametr integer
↳(tam ədəd tipi) tipində ədədləri-argument olaraq alır.Qutunun ifadə sayını 8
↳sayda tənzimləyək
```

```
[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↳kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setAlignment(Qt.AlignLeft)#kursoru solda göstərir
        self.lineedit.setMaxLength(8)
        self.lineedit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

```
[3]: #setReadOnly() metodu
#Metod iki parametr alır; True və False
#Qutuda olan ifadənin sadəcə görünməsini təşkil etmək üçün metoda True
↳ifadəsini veririk.
```

```
[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser
```



```

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı_
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setText(u'Mağın Öyrənmə')
        self.lineedit.setReadOnly(True)
        self.lineedit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: #text() parametri

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı_
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()

```

```

        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setText(u'Maşın Öyrənmə')
        text = self.lineedit.text()
        print('Qutuda olan mətn:{}'.format(text))
        self.lineedit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Qutuda olan mətn:Maşın Öyrənmə

[3]: *#setValidator() metodu*  
*#Metod vasitəsilə qutu daxilində olan elementlərin ədəd tipini tərtib edə*  
*↳bilərik.*  
*#1.QIntValidator -tam ədədlər üçün*  
*#2. QDoubleValidator -Həqiqi(float) ədədlər üçün*

[ ]: *# kitabxanaları proqrama daxil edirik*

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↳kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.setValidator(QDoubleValidator())
        text = self.lineedit.text()

```

```

        self.lineEdit.move(200,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[3]: *#setInputMask() metodu*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.label = QLabel(self)
        self.label.setText(u'Adınız')
        self.label.move(150,50)

        self.lineEdit = QLineEdit(self)
        self.lineEdit.setAlignment(Qt.AlignLeft)
        self.lineEdit.move(200, 50)

        self.label1 = QLabel(self)
        self.label1.setText((u'Yaşınız'))
        self.label1.move(150,80)
        self.lineEdit1 = QLineEdit(self)
        self.lineEdit1.setInputMask('99')
        self.lineEdit1.setAlignment(Qt.AlignLeft)
        self.lineEdit1.move(200,80)
        self.button = QPushButton(self)

```

```

        self.button.setText(u'click')
        self.button.move(230,100)
        self.button.clicked.connect(self.goster)
        self.show()
    def goster(self):
        print('Adım :{} | Yaşım :{}'.format(self.lineedit.text(),self.lineedit1.
↪text()))
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

Adım :Rəşad | Yaşım :33
Adım :Rəşad | Yaşım :33
Adım :Rəşad | Yaşım :33
Adım :Rəşad | Yaşım :33

```

[2]: *#lavə olaraq QPushButton sinifi istifadə etdik.Növbəti dərslərdə bu klası*  
↪*ətraflı öyrənəcəyik*

[4]: *#setFont*

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.font = QFont('Image preprocessing',12,QFont.Bold)
        self.lineedit.setFont(QFont(self.font))
        self.show()

```

```

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: *#editingFinished() metodu*

[2]: *# kitabxanaları proqrama daxil edirik*

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineEdit = QLineEdit(self)
        self.lineEdit.editingFinished.connect(self.showme)
        self.lineEdit.move(140,50)
        self.show()
        def showme(self):
            print(u'Düyməyə basıldı')
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Düyməyə basıldı

[3]: *#returnPressed() metodu*

[ ]: *# kitabxanaları proqrama daxil edirik*

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *

```

```

import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı_
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.returnPressed.connect(self.webbrowser)
        self.lineedit.move(140,50)
        self.show()

    def webbrowser(self):
        webbrowser.open('http://www.'+self.lineedit.text())

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: #selectionChanged() metodu

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı_
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()

```



[illegible]

[4]: #Metod vasitəsiylə qutuda ifadə yazıldığı an hərəkət alır.Yəni biz alternativ  
↪funksiyalarla istifadəçiyə mesaj ötürə bilərik

33

```

import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪ küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.textChanged.connect(self.change)
        self.lineedit.move(140,50)
        self.show()

    def change(self):
        print('Mətn dəyişdirildi')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Mətn dəyişdirildi  
 Mətn dəyişdirildi  
 Mətn dəyişdirildi  
 Mətn dəyişdirildi

[2]: *#textEdited() metodu*

[3]: *#Qutuda mətn dəyişdirildiyi an hərəkət alır*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)

```

```

        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.textEdited.connect(self.changed)#mətni qutudan sildikdə
↪belə aktiv olur
        self.lineedit.move(140,50)
        self.show()
        def changed(self):
            print('Mətn seçildi')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Mətn seçildi  
 Mətn seçildi  
 Mətn seçildi  
 Mətn seçildi  
 Mətn seçildi  
 Mətn seçildi

[2]: #cursorPositionChanged() metodu

[3]: #Metod maus və kursora reaksiya verir.

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu

```

```

        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.lineedit = QLineEdit(self)
        self.lineedit.cursorPositionChanged.connect(self.changed)
        self.lineedit.move(140,50)
        self.show()
    def changed(self):
        print('Siz qutuya toxundunuz')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Siz qutuya toxundunuz  
 Siz qutuya toxundunuz  
 Siz qutuya toxundunuz  
 Siz qutuya toxundunuz

[ ]:

[ ]:

### 0.1.1 QPushButton

buton-düymə.Siz bu düymələrə gün ərzində tez-tez rast gəlməlisiniz.proqramlarda,evdə elektrik cərəyanı , dövrə açarını bağlamaq üçün və sairə

Bu düymələrin hər birinin özəl təyinatı olur. laqəli vəzifələri yerinə yetirirlər.Eynilə bu bəhisdə də biz bu düymələrin basıldığı anda hazırladığımız təyinatı funksiyalarla əlaqələndirib,müəyyən kodlarımızı icra etdirəcəyik.

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.

```

```

        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Başla')
        self.button.move(220,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Qeyd edimki əvvəlki bəhlərdə keçirilən metodların bir neçəsi ümumi metodlardır.və digər siniflərlə də işlədilər bilinir.Məsələn yuxarıda gördüyünüz setText metodunu QLabel,QLineEdit siniflərində də istifadə etmişdik.

```
[2]: # setIcon()
```

```
[3]: #Metod vasitəsilə düymələrə rəsm əlavə edə bilərik
```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)

```

```

        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Başla')
        self.button.setIcon(QIcon('../image/bstart.png'))
        self.button.move(220,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Kodlarımızda düymə üçün həm yazı həm də rəsm hazırladıq.

Və rəsm ölçüsünü də tənzimləyə bilərik

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Başla')
        self.button.setIcon(QIcon('../image/bstart.png'))
        self.button.setIconSize(QSize(25,25))
        self.button.move(220,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Hal-hazırda hazırladığımız düymə fəaliyyətsizdir. vətəli bəhslərdə connect ifadəsini istifadə etmişdik.verilən parametri digər parametr ilə əlaqələndirir.

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, time
var1 =time.time()
class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.

        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Başla')
        self.button.setIcon(QIcon('../image/bstart.png'))
        self.button.move(220,50)
        self.button.clicked.connect(self.Speed)
        self.show()
    def Speed(self):
        var2 = time.time()
        delta = var1-var2
        self.label = QLabel(self)
        self.label.setText('{} saniyə'.format(str(delta)))
        print('Proqramın başladığı andan,düymənin çalışdığı ana qədər vaxtın
        ➔hesablanması:{}'.format(delta))
        self.label.move(220,80)
        self.label.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

Proqramın başladığı andan,düymənin çalışdığı ana qədər vaxtın  
hesablanması:-8.838595390319824

Biz bu vaxt anlayışları ilə processing qutularının real artımı proseslərini də həyata keçirəcəyik. Hal - hazırda PyQt -sinif və metodlarını nümunələrlə öyrənirik. İrəliləyən dərslərdə daha irəli gedib brauzer vebcam a bağlanması və s proqramlar hazırlayacağıq.

[2]: *#setCheckable() metodu*

Metod susmaya görə həmişə True ifadəsində qalır. boolean tipində iki dəyər, True və False ifadələrini alır. Məsələn siz proqrama daxil olursunuz. Proqram sizdən ardıcıl qutularda adınızı soyadınızı və şifrənizi soruşur. Xanalar boş olduqda düymə aktiv olmur. n son şifrə qutusu doldurduqda isə düymə aktivləşir. Bu kimi nümunələrə tez-tez rast gəlinirsiniz. Bu tipli proqramlar hazırladıqda metod- dan istifadə edə bilərsiniz

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə logosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Başla')
        self.button.setIcon(QIcon('../image/bstart.png'))
        self.button.move(230, 60)
        self.button.setEnabled(False)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

[2]: *#Düymə daxilində 'Drop down' menu*

Növbəti dərslərdə menu -hazırlanmasını ətraflı öyrənəcəyik. Hal-hazırda nümunə üçün göstərirəm



```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Menu')
        self.button.move(230, 60)
        self.menu = QMenu(self)
        self.menu1 = self.menu.addAction('Open')
        self.menu2 = self.menu.addAction('Quit')
        self.menu3 = self.menu.addAction('Save')
        self.button.setMenu(self.menu)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

### 0.1.2 Növbəti istifadə edəcəyimiz parametrlər

```
[3]: #setToolTip() metodu
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):
```

```

def __init__(self, parent=None):
    super(MainWindow, self).__init__(parent)
    # sas pəncərə self ifadəsində referans alındı.
    self.setWindowTitle("Bu mənim ilk proqramım")
    # Pəncərə loqosu
    self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
    ➔ küncdə loqo əlavə olundu
    # Pəncərə arxa plan rəngi
    self.palette = QPalette()
    self.palette.setColor(QPalette.Background, Qt.white)
    self.setPalette(self.palette)
    self.setGeometry(600,150,512,512)
    self.button = QPushButton(self)
    self.button.setText(u'Oxu')
    self.button.move(230, 60)
    self.button.setToolTip(u'Düymə haqqında ətraflı məlumat')
    self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Düymənin üzərinə gəldikdə əlavə etdiyimiz məlumat mesajı görünür

[2]: `#setStyleSheet`

Css atributlarından istifadə edərək yazı tipi,ölçüləri rənglərini dəyişə bilərik

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔ küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)

```

```

        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.button = QPushButton(self)
        self.button.setText(u'Oxu')
        self.button.move(230, 60)
        self.button.setToolTip(u' <b>məlumat pəncərəsi</b> ')
        self.button.setStyleSheet('QToolTip { font-size: 8pt; font-family:
↪Sansserif;}')
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[ ]:

[2]: *#QRadioButton class*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.radiobutton = QRadioButton('PyQt dərsləri',self)
        self.radiobutton.move(220,50)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```
[2]: #isChecked()
```

metod vasitəsilə seçilmiş radio düyməsi - funksiya ilə əlaqələndirilib məlumat verə bilər

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self.radioButton = QRadioButton('Windows', self)
        self.radioButton.move(2, 2)
        self.radioButton.clicked.connect(self.click)
        self.radioButton1 = QRadioButton('Ubuntu', self)
        self.radioButton1.move(2, 22)
        self.radioButton1.clicked.connect(self.click)
        self.radioButton2 = QRadioButton('Mac Os', self)
        self.radioButton2.move(2, 42)
        self.radioButton2.clicked.connect(self.click)
        self.show()

    def click(self):
        if self.radioButton.isChecked():
            print('Windows')
        elif self.radioButton1.isChecked():
            print('Ubuntu')
        elif self.radioButton2.isChecked():
            print('Mac Os')
        else:
            print('Nothing selected')

if __name__ == '__main__':
```

```

app = QApplication([])
gui = MainWindow()
app.exec_()

```

Windows  
 Ubuntu  
 Mac Os  
 Windows

```
[2]: #text()
```

metod vasitəsilə seçilmiş radio düyməsinə əlavə edilmiş mətni ala bilərik

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(600,150,512,512)
        self radiobutton = QRadioButton('Windows', self)
        self radiobutton.move(2, 2)
        self radiobutton.clicked.connect(self.click)
        self radiobutton1 = QRadioButton('Ubuntu', self)
        self radiobutton1.move(2, 22)
        self radiobutton1.clicked.connect(self.click)
        self radiobutton2 = QRadioButton('Mac Os', self)
        self radiobutton2.move(2, 42)
        self radiobutton2.clicked.connect(self.click)
        self.show()

    def click(self):
        if self radiobutton.isChecked():
            print(self radiobutton.text())

```

```

        elif self.radioButton1.isChecked():
            print(self.radioButton1.text())
        elif self.radioButton2.isChecked():
            print(self.radioButton2.text())
        else:
            print('Nothing selected')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Mac Os  
 Ubuntu  
 Windows  
 Ubuntu

[2]: *#setChecked metodu*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪ küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.radioButton = QRadioButton('Windows', self)
        self.radioButton.move(2, 2)
        self.radioButton.clicked.connect(self.click)
        self.radioButton1 = QRadioButton('Ubuntu', self)
        self.radioButton1.move(2, 22)
        self.radioButton1.clicked.connect(self.click)
        self.show()

```

```

def click(self):
    if self.radiobutton.isChecked():
        self.radiobutton.setText(u'Windows əməliyyat sistemi')
        print(self.radiobutton.text())

    elif self.radiobutton1.isChecked():
        self.radiobutton1.setText(u'Ubuntu əməliyyat sistemi')
        print(self.radiobutton1.text())
    else:
        print('Nothing selected')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

Ubuntu əməliyyat sistemi
Windows əməliyyat sistemi
Ubuntu əməliyyat sistemi

```

[ ]:

[3]: #QCheckBox

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.checkbox1 = QCheckBox('Tkinter', self)

```

```

        self.checkbox1.move(20, 20)
        self.checkbox2 = QCheckBox('PyQt', self)
        self.checkbox2.move(20, 0)
        self.checkbox3 = QCheckBox('wxPython', self)
        self.checkbox3.move(20, 40)
        self.checkbox3.setTristate(False)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: *#checked və unchecked*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.checkbox1 = QCheckBox('Tkinter', self)
        self.checkbox1.move(20, 0)
        self.checkbox1.stateChanged.connect(self.check)
        self.checkbox2 = QCheckBox('PyQt', self)
        self.checkbox2.move(20, 20)
        self.checkbox2.stateChanged.connect(self.check)
        self.show()

    def check(self):
        if self.checkbox1.text() == 'Tkinter':
            if self.checkbox1.isChecked() == True:
                print(self.checkbox1.text())

```



```

        if self.checkbox2.text() == 'PyQt' :
            if self.checkbox2.isChecked() == True:
                print(self.checkbox2.text())

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Tkinter  
 PyQt  
 PyQt  
 Tkinter  
 Tkinter  
 Tkinter  
 PyQt  
 PyQt

[2]: *#setTristate*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪ küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.checkbox1 = QCheckBox('Tkinter', self)
        self.checkbox1.move(20, 0)
        self.checkbox1.stateChanged.connect(self.check)
        self.checkbox2 = QCheckBox('PyQt', self)
        self.checkbox2.move(20, 20)
        self.checkbox2.stateChanged.connect(self.check)
        self.checkbox2.setTristate(True)

```

```

        self.show()
    def check(self):
        if self.checkbox1.text() == 'Tkinter':
            if self.checkbox1.isChecked() == True:
                print(self.checkbox1.text())
        if self.checkbox2.text() == 'PyQt' :
            if self.checkbox2.isChecked() == True:
                print(self.checkbox2.text())

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

PyQt  
PyQt  
Tkinter  
PyQt  
Tkinter  
PyQt  
Tkinter  
PyQt  
Tkinter  
Tkinter

[ ]:

### 0.1.3 QComboBox Class

Qutu daxilində ifadələri siyahı formada göstərir.

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi

```

```

        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        self.combobox.addItem('Python')
        self.combobox.addItem('Java')
        # və ya siyahı tərtib edib
        # siyahı = ['Python', 'Java', 'C++', 'Perl']
        # self.combobox.addItem(siyahı)
        self.combobox.move(120,50)

        self.show()
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: *#clear() addItem addItems*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        siyahı = ['Python', 'Java', 'C++', 'Perl']
        self.combobox.addItem(siyahı)

```

```

        self.combobox.move(120,50)
        self.button = QPushButton('delete', self)
        self.button.move(120, 80)
        self.button1 = QPushButton('add', self)
        self.button1.move(210, 80)
        self.button.clicked.connect(self.clean)
        self.button1.clicked.connect(self.add)
        self.show()
    def clean(self):
        self.combobox.clear()
    def add(self):
        ai = ['AI', 'ML', 'DL']
        self.combobox.clear()
        self.combobox.addItem(ai)
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[3]: *#currentIndexChanged()*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        siyahı = ['Python', 'Java', 'C++', 'Perl']
        self.combobox.addItem(siyahı)
        self.combobox.move(120,50)

```

```

        self.combobox.currentIndexChanged.connect(self.choose)
        self.show()
    def choose(self):
        print('Qutuda ümumi {} dil var'.format(self.combobox.count()))
        print(u'Siz, indeks nömrəsi {} olan {} dili seçdiniz'.format(self.
↪combobox.currentIndex(), self.combobox.currentText()))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Qutuda ümumi 4 dil var  
 Siz, indeks nömrəsi 1 olan Java dili seçdiniz  
 Qutuda ümumi 4 dil var  
 Siz, indeks nömrəsi 2 olan C++ dili seçdiniz  
 Qutuda ümumi 4 dil var  
 Siz, indeks nömrəsi 0 olan Python dili seçdiniz

[2]: *#highlighted*

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə logosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        siyahı = ['Python', 'Java', 'C++', 'Perl']
        self.combobox.addItem(siyahı)
        self.combobox.move(120,50)

```

```

        self.combobox.highlighted.connect(self.choose)
        self.show()
    def choose(self, index):
        if index == self.combobox.currentIndex():
            print(u'Siz {} dili seçdiniz'.format(self.combobox.currentText()))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Siz Python dili seçdiniz  
 Siz Java dili seçdiniz  
 Siz C++ dili seçdiniz

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        siyahı = ['Python', 'Java', 'C++', 'Perl']
        self.combobox.addItem(sorted(siyahı))
        self.combobox.move(120,50)
        self.combobox.highlighted.connect(self.choose)
        self.combobox.setStyleSheet('QComboBox{background-color: darkgray;
        ↪selection-background-color: gray;color: black;}')
        self.show()
    def choose(self, index):
        if index == self.combobox.currentIndex():

```

```

        print(u'Siz {} dili seçdiniz'.format(self.combobox.currentText()))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Siz C++ dili seçdiniz  
 Siz Java dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz  
 Siz Perl dili seçdiniz

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setStyleSheet("background-color:blue")
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.label = QLabel('Combobox',self)
        self.label.move(50,50)
        self.combobox = QComboBox(self)
        siyahı = ['Python','Java','C++','Perl']
        self.combobox.addItem(sorted(siyahı))
        self.combobox.move(120,50)
        self.combobox.highlighted.connect(self.choose)
        self.combobox.setStyleSheet('QComboBox{background-color: darkgray;
        ↪selection-background-color: gray;color: black;}')
        self.show()

```

```

def choose(self,index):
    if index == self.combobox.currentIndex():
        print(u'Siz {} dili seçdiniz'.format(self.combobox.currentText()))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: *#QBoxLayout Class*

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.button = QPushButton('vbox', self)
        self.button1 = QPushButton('vbox1', self)
        self.vbox = QVBoxLayout() # vertikal istiqamətdə V-ifadəsi
        self.vbox.addWidget(self.button)
        self.vbox.addWidget(self.button1)
        self.setLayout(self.vbox)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```



```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.button = QPushButton('hbox', self)
        self.button1 = QPushButton('hbox1', self)
        self.hbox = QHBoxLayout() # horizontal istiqamətdə H -ifadəsi
        self.hbox.addWidget(self.button)
        self.hbox.addWidget(self.button1)
        self.setLayout(self.hbox)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

[2]: #addStretch() metodu ,düymələr arasında məsafə yaradır.Və bu məsafə,pəncərənin  
 ↪dinamik ölçüsünü dəyişdikdə düymələr arasında məsafələr də dəyişir.

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
```

```

        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.button = QPushButton('hbox', self)
        self.button1 = QPushButton('hbox1', self)
        self.hbox = QHBoxLayout() # vertikal istiqamətdə V -ifadəsi
        self.hbox.addWidget(self.button)
        self.hbox.addStretch()
        self.hbox.addWidget(self.button1)
        self.setLayout(self.hbox)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: #QGridLayout Class -klasi  
#sinif, sətir və sütunlarla verilənləri yerləşdirir.

[1]: # kitabxanaları proqrama daxil edirik

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,400,400)
        self.grid = QGridLayout(self)

```

```

        count = 1
        for i in range(3):
            for v in range(1, 6):
                self.grid.addWidget(QPushButton(str(count),self),i,v)#3 sətir 5
↪ sütun
                count+=1
            self.setLayout(self.grid)
            self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Calculate")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪ küncdə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.grid = QGridLayout(self)
        self.setLayout(self.grid)
        self.lineedit = QLineEdit(self)

        list = [ '1', '2', '3', '/', '<--', '4', '5', '6', '*', 'C', '7', '8',
↪ '9', '+', '-', '0', '.', '=', '-', 'Quit']
        post = [(i, j) for i in range(5) for j in range(5)]
        for show, number in zip(post, list):
            if number == '':
                continue
            self.grid.addWidget(self.lineedit, 0, 5)
            self.button = QPushButton(number,self)
            self.grid.addWidget(self.button, *show)
        self.show()

```

```

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: *#Sadə hesablayıcı program hazırlayaq*

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.grid = QGridLayout()
        self.setLayout(self.grid)
        self.line = QLineEdit()
        self.line.setMaxLength(17)
        self.line.setAlignment(Qt.AlignRight)
        # button -----
        button_plus = QPushButton('+')
        button_minus = QPushButton('-')
        button_mult = QPushButton('*')
        button_div = QPushButton('/')
        button_equal = QPushButton('=')
        button_equal.clicked.connect(self.result)
        button_clear = QPushButton('CE')
        button_clear.clicked.connect(self.clean)
        button_backsps = QPushButton('Backspace')
        button_backsps.clicked.connect(self.back)
        button_exit = QPushButton('Quit')
        button_exit.clicked.connect(self.quiting)
        list_operator = [ \
            button_plus, button_minus,

```

```

        button_mult, button_div, ]
    for i in list_operator:
        i.clicked.connect(self.opert)

    list = [ \
        '1', '2', '3',
        '4', '5', '6',
        '7', '8', '9',
        '0', '.']
    post = [(i, j) for i in range(5) for j in range(5)]
    for show, number in zip(post, list):
        if number == '':
            continue
        # QGridLayout -----
        self.grid.addWidget(self.line, 0, 5)
        self.button = QPushButton(number)
        self.button.clicked.connect(self.operation)
        self.grid.addWidget(self.button, *show)
        self.grid.addWidget(button_plus, 2, 1)
        self.grid.addWidget(button_minus, 2, 2)
        self.grid.addWidget(button_mult, 2, 3)
        self.grid.addWidget(button_div, 2, 4)
        self.grid.addWidget(button_equal, 3, 4)
        self.grid.addWidget(button_clear, 1, 5)
        self.grid.addWidget(button_backsps, 2, 5)
        self.grid.addWidget(button_exit, 3, 5)
    self.show()
    # all main function-----
def result(self):
    try:
        i = self.line.text()
        self.line.clear()
        self.line.setText(str(eval(str(i))))
    except NameError:
        self.line.clear()
        self.line.setText('please write only integer')
    except SyntaxError:
        self.line.clear()
    except ZeroDivisionError:
        self.line.setText('0')
def operation(self):
    sender = self.sender()
    sender.text()
    if False == False:
        self.line.setText(self.line.text() + sender.text())
    else:
        self.line.setText(sender.text())

```

```

def opert(self):
    sender = self.sender()
    i = sender.text()
    if False == False:
        self.line.setText(self.line.text() + i)
def clean(self):
    self.line.clear()
def back(self):
    self.line.backspace()
def quitting(self):
    self.close()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```
[1]: #QFormLayout Class
```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.label = QLabel('Name')
        self.lineedit = QLineEdit(self)
        self.form = QFormLayout(self)
        self.form.addRow(self.label, self.lineedit)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()

```

```
app.exec_()
```

```
[1]: # kitabxanalari proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, webbrowser

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.label = QLabel('Name')
        self.lineedit = QLineEdit(self)
        self.form = QFormLayout(self)
        self.form.addRow(self.label, self.lineedit)
        self.vbox = QVBoxLayout(self)
        self.vbox.addWidget(self.label)
        self.vbox.addWidget(self.lineedit)
        self.form.addRow(self.vbox)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

```
[1]: # kitabxanalari proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
```

```

        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.label = QLabel('Name')
        self.label1 = QLabel('Surname')
        self.lineedit = QLineEdit(self)
        self.lineedit1 = QLineEdit(self)
        self.form = QFormLayout(self)
        self.vbox = QVBoxLayout(self)
        self.tobox = QVBoxLayout(self)
        self.vbox.addWidget(self.label)
        self.vbox.addWidget(self.lineedit)
        self.tobox.addWidget(self.label1)
        self.tobox.addWidget(self.lineedit1)
        self.form.addRow(self.vbox)
        self.form.addRow(self.tobox)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: #QSpinBox

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()

```



```

        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.spinbox = QSpinBox(self)
        self.spinbox.resize(200, 40)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.spinbox = QSpinBox(self)
        self.spinbox.resize(200, 40)
        self.spinbox.setMinimum(-10)
        self.spinbox.setMaximum(30)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[2]: #setRange(int,int) metodu
#Metod,minimal və maksimal dəyərləri mötərizə daxilində bərabər qəbul edir.Yəni
➔yuxarıdakı metodların əvəzinə birbaşa bu metodu istifadə edə bilərik.

```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *

```

```

from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.spinbox = QSpinBox(self)
        self.spinbox.resize(200, 40)
        self.spinbox.setRange(-5, 10)
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ➔kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.spinbox = QSpinBox()

```

```

        self.label = QLabel('Spinbox value:')
        self.spinbox.resize(100, 25)
        self.spinbox.setRange(1994, 2009)
        self.spinbox.setValue(200)
        self.grid = QGridLayout(self)
        self.grid.addWidget(self.spinbox)
        self.grid.addWidget(self.label)
        self.spinbox.valueChanged.connect(self.create)
        self.show()

    def create(self):
        self.label.setText('Value:' + str(self.spinbox.value()))
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```
[1]: #QSlider
```

```
[2]: #QSlider(Qt.Horizontal)
      #QSlider(Qt.Vertical)
```

```
[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)
        self.label = QLabel(self)
        self.label.setPixmap(QPixmap('../image/ai.jpg'))
        self.slider = QSlider(self)
        self.slider.setGeometry(50, 50, 20, 200)
        self.slider.setRange(0, 200)

```

```

        self.slider.setTickPosition(QSlider.TicksLeft)
        self.slider.setFocusPolicy(Qt.NoFocus)
        self.slider.valueChanged.connect(self.changed)
        self.show()
    def changed(self,value):
        if value == 0:
            self.label.setGeometry(160,50,80,50)
        elif value > 0 and value <= 100:
            self.label.setGeometry(160, 50, 100, 100)
        elif value >= 100 and value <= 200:
            self.label.setGeometry(160, 50, 200, 200)

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: #valueChanged()
      #sliderPressed() Signal-pressed
      #sliderMoved()
      #sliderReleased()

```

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.label = QLabel('QSlider', self)
        self.label.move(150, 20)
        self.slider = QSlider(Qt.Horizontal, self)
        self.slider.move(150, 60)
        self.slider.setRange(0, 30)

```

```

        self.slider.setTickPosition(QSlider.TicksLeft)
        self.slider.setTickInterval(5)
        self.slider.setValue(20)
        self.slider.valueChanged.connect(self.change)
        self.show()

    def change(self):
        self.label.setFont(QFont('Italic', self.slider.value()))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)
        self.label = QLabel('QSlider', self)
        self.label.move(150, 20)
        self.slider = QSlider(Qt.Horizontal, self)
        self.slider.move(150, 60)
        self.slider.setRange(0, 30)
        self.slider.setTickPosition(QSlider.TicksLeft)
        self.slider.setTickInterval(5)
        self.slider.setValue(20)
        self.slider.sliderMoved.connect(self.change)
        self.slider.sliderPressed.connect(self.change1)
        self.show()

    def change(self):

```

```

        print('Moved')
    def change1(self):
        print('Pressed')

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

Pressed  
 Pressed  
 Moved  
 Moved  
 Moved  
 Pressed  
 Moved  
 Moved  
 Moved  
 Moved  
 Moved

[2]: #QSplash

[3]:

```

#1.Qt.AlignCenter
#2.Qt.AlignBottom
#3.Qt.AlignHCenter (Horizontal istiqamətdə mərkəzdə)
#4.Qt.AlignVCenter (Vertical istiqamətdə mərkəzdə)
#5.Qt.AlignLeft
#6.Qt.AlignRight

```

[ ]:

```

# kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()

```

```

        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)
        self.splash = QSplashScreen(QPixmap('../image/ai.png'))
        self.splash.show()
        self.loading(self.splash)
        self.splash.finish(self)
        self.show()
    def loading(self, s):
        for i in range(1, 11):
            time.sleep(2)
            s.showMessage(u'Proqram yüklənir, zəhmət olmasa gözləyin...{ }% '.
↪format(i*10), Qt.AlignCenter, Qt.black)

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: #QMenuBar, QMenu

[3]: # kitabxanaları proqrama daxil edirik

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, time

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)
        self.menu = QMenuBar(self)
        self.menu.addMenu('File')
        self.menu.addMenu('Edit')
        self.show()

if __name__ == '__main__':

```

```
app = QApplication([])
gui = MainWindow()
app.exec_()
```

[4]: *#For QMainWindow*

[3]: *#menuBar() - klassik menu çubuğu*  
*#addMenu - menu çubuğuna əlavələr*  
*#addAction - alt menuların yerləşdirilməsi (eləcə də icon və mətnlər üçün)*  
*#setEnabled(True or False) - menu çubuğunda olan əlavələri aktiv və passiv*  
*→ etmək.*  
*#addSeparator() - alt menuları bir xətti ayırmaq*  
*#clear() - menu çubuğunu götürmək, silmək*  
*#setShortcut - menu-nu klaviatura ilə əlaqələndirmək*  
*#setText() - ifadə əlavə etmək*  
*#text() - ifadələri götürmək*  
*#title() - QMenu üçün başlıq*

[1]: *# kitabxanaları proqrama daxil edirik*

```
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        →kündə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)
        menubar = self.menuBar()
        f = menubar.addAction('&File')
        f.addAction('New')
        f.addAction('Open')
        f.addAction('Save')
        f.addAction('Close')
        e = menubar.addAction('&Edit')
        e.addAction('view')
        self.show()
```



```

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[4]: *#Menu çubuğunu ümumilikdə css atributları ilə dizayn etmək üçün*

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        menubar = self.menuBar()
        menubar.setStyleSheet("color: #365471;"
                               "background-color: black;"
                               "selection-color: #172533;"
                               "selection-background-color: #BFBFBF;")

        f = menubar.addMenu('File')
        f.addAction('New')
        f.addAction('Open')
        f.addAction('Save')
        f.addAction('Close')
        e = menubar.addMenu('Edit')
        e.addAction('view')
        self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: *# kitabxanaları proqrama daxil edirik*

```

from PyQt5.QtWidgets import *

```

```

from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        menubar = self.menuBar()
        f = menubar.addMenu('File')
        f.addAction('New')
        f.addAction('Open')
        f.addAction('Save')
        close = QAction('Close', self)
        close.setShortcut("Esc")
        close.triggered.connect(self.closing)
        f.addAction(close)
        e = menubar.addMenu('Edit')
        e.addAction('view')
        self.show()

    def closing(self):
        self.close()
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):

```

```

super(MainWindow, self).__init__(parent)
# sas pəncərə self ifadəsində referans alındı.
self.setWindowTitle("Bu mənim ilk proqramım")
# Pəncərə loqosu
self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↳ küncdə logo əlavə olundu
# Pəncərə arxa plan rəngi
self.palette = QPalette()
self.palette.setColor(QPalette.Background, Qt.white)
self.setPalette(self.palette)
self.setGeometry(150, 50, 512, 512)
menubar = self.menuBar()
f = menubar.addMenu('File')
f.setStyleSheet(" background-color:gray;"
                 "border-style: outset;"
                 "border-width: 1px;"
                 "border-radius: 10px;"
                 "border-color: green;"
                 "font: bold 12px;"
                 "min-width: 10em;")

f.addAction('New')
f.addAction('Open')
f.addAction('Save')
close = QAction('Close', self)
close.setShortcut("Esc")
close.triggered.connect(self.closing)
f.addAction(close)
e = menubar.addMenu('Edit')
e.addAction('view')
self.show()

def closing(self):
    self.close()
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: #QToolBar

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

```

```

def __init__(self, parent=None):
    super(MainWindow, self).__init__(parent)
    # sas pəncərə self ifadəsində referans alındı.
    self.setWindowTitle("Bu mənim ilk proqramım")
    # Pəncərə loqosu
    self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
    ↪ küncdə loqo əlavə olundu
    # Pəncərə arxa plan rəngi
    self.palette = QPalette()
    self.palette.setColor(QPalette.Background, Qt.white)
    self.setPalette(self.palette)
    self.setGeometry(150, 50, 512, 512)
    self.toolbar = self.addToolBar('')
    self.new = QAction(QIcon('image/new.jpg'), 'New', self)
    self.open = QAction(QIcon('image/open.png'), 'Open', self)
    self.exit = QAction(QIcon('image/exit.png'), 'Exit', self)
    self.toolbar.addAction(self.new)
    self.toolbar.addAction(self.open)
    self.toolbar.addAction(self.exit)
    self.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys, time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
    ↪ küncdə loqo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512, 512)

```

```

self.toolbar = self.addToolBar('')
self.new = QAction(QIcon('image/new.jpg'), 'New', self)
self.open = QAction(QIcon('image/open.png'), 'Open', self)
self.exit = QAction(QIcon('image/exit.png'), 'Exit', self)
self.exit.triggered.connect(self.closed)
self.toolbar.addAction(self.new)
self.toolbar.addAction(self.open)
self.toolbar.addAction(self.exit)
self.show()
def closed(self):
    self.close()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.toolbar = self.addToolBar('')
        self.toolbar.setIconSize(QSize(15,15))
        self.new = QAction(QIcon('image/new.jpg'), 'New', self)
        self.open = QAction(QIcon('image/open.png'), 'Open', self)
        self.exit = QAction(QIcon('image/exit.png'), 'Exit', self)
        self.exit.triggered.connect(self.closed)
        self.toolbar.addAction(self.new)
        self.toolbar.addAction(self.open)
        self.toolbar.addAction(self.exit)

```

```

        self.toolbar.setToolButtonStyle(Qt.ToolButtonTextBesideIcon|Qt.
↪AlignLeading)
        self.lineedit = QLineEdit(self)
        self.toolbar.addWidget(self.lineedit)
        self.toolbar.setMovable(False)
        self.show()
    def closed(self):
        self.close()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: #QDialog Class

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.toolbar = self.addToolBar('')
        self.toolbar.setIconSize(QSize(15,15))
        self.new = QAction(QIcon('image/new.jpg'),'New',self)
        self.open = QAction(QIcon('image/open.png'),'Open',self)
        self.exit = QAction(QIcon('image/exit.png'), 'Exit', self)
        self.toolbar.addAction(self.new)
        self.toolbar.addAction(self.open)
        self.toolbar.addAction(self.exit)
        self.new.triggered.connect(self.dialog)
        self.show()
    def dialog(self):

```

```

self.dialog = QDialog()
self.button = QPushButton('Ok',self.dialog)
self.button.move(50,50)
self.button.clicked.connect(self.dialog.accept)
self.dialog.show()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[2]: #QMessageBox

[3]: #QMessageBox

```

#Məlumat pəncərəsidir.Ümumi dörd formada məlumat verir.
#Sual Question(QMessageBox.question)
#Məlumat Information( QMessageBox.information)
#Həyəcan Warning( QMessageBox.warning)
#Kəskin həyəcan(kritik)
#Critical( QMessageBox.critical)
#Haqqında About ( QMessageBox.about) icon -no..
#QMessageBox düymə formatları
#QMessageBox.Ok QMessageBox.Close
#QMessageBox.Open QMessageBox.Yes
#QMessageBox.Save QMessageBox.No
#QMessageBox.Cancel QMessageBox.Abort
#QMessageBox.Retry QMessageBox.Ignore

```

[1]: # kitabxanaları proqrama daxil edirik

```

from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)

```

```

        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.button = QPushButton('exit',self)
        self.button.move(100,60)
        self.button.clicked.connect(self.mesaj)
        self.show()
    def mesaj(self):
        self.mesaj = QMessageBox()
        self.mesaj.setWindowTitle('Mesaj pəncərəsi')
        self.mesaj = QMessageBox.question(self, 'Title', 'Are you sure quit?'
↪',QMessageBox.Yes | QMessageBox.No)
        if self.mesaj == QMessageBox.Yes:
            self.close()

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[ ]:

[3]: #QInputDialog

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys,time

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı↵
↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.button = QPushButton('Dialog',self)
        self.button.clicked.connect(self.dialog)
        self.setFocus()
        self.lineedit = QLineEdit(self)

```



```

        self.lineedit.move(130,0)
        self.show()
    def dialog(self):
        text, ok = QInputDialog.getText(self, 'Input Dialog', 'Enter you name')
        if ok:
            self.line.setText(unicode(text))

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

[1]: `#QFontDialog`

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(150, 50, 512,512)
        self.label = QLabel(u'Python ', self)
        self.label.move(150, 20)
        self.button = QPushButton('Font', self)
        self.button.move(140, 80)
        self.button.clicked.connect(self.q)
        self.show()

    def q(self):
        okey, font = QFontDialog.getFont()
        if font:
            self.label.setFont(okey)

if __name__ == '__main__':
    app = QApplication([])

```

```
gui = MainWindow()
app.exec_()
```

```
[3]: #QFileDialog
```

```
[ ]: # kitabxanaları programa daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə logosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,600,600)
        self.button = QPushButton('Open file', self)
        self.button.move(20, 20)
        self.button.clicked.connect(self.getfile)
        self.label = QLabel(self)

        self.show()

    def getfile(self):
        f = QFileDialog.getOpenFileName(self, 'Openfile', '../image/', '*.jpg')
        imagePath = f[0]
        pixmap = QPixmap(imagePath)
        self.label.setPixmap(QPixmap(pixmap))
        self.label.setGeometry(10, 50, pixmap.width(),pixmap.height())
        self.resize(pixmap.width(),pixmap.height())

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()
```

```

[2]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QMainWindow):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk programım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,600,600)
        self.button = QPushButton('Open file', self)
        self.button.move(20, 20)
        self.button.clicked.connect(self.getfile)
        self.textedit = QTextEdit(self)
        self.textedit.setGeometry(30, 100, 550, 400)

        self.show()

    def getfile(self):
        f = QFileDialog.getOpenFileName(self, 'Openfile')
        file = open(f[0], 'r', encoding='utf8')
        with file:
            text = file.read()
            self.textedit.setText(text)

if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[3]: #QFileDialog.getSaveFileName

```

```

[4]: print("""

```

```

def save_file(self):
    self.name=QFileDialog.getSaveFileName(self,'Open file','/home','Save File(*.
    ↪py)')
    file=open(self.name,'w')
    text=self.textedit.toPlainText()
    file.write(text)
    file.close

    """

```

```

def save_file(self):
    self.name=QFileDialog.getSaveFileName(self,'Open file','/home','Save
File(*.py)')
    file=open(self.name,'w')
    text=self.textedit.toPlainText()
    file.write(text)
    file.close

```

```
[5]: #QTabWidget()
```

```

[ ]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə logosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪küncdə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)

```

```

        self.setGeometry(400,400,600,600)
        self.tab = QTabWidget(self)
        self.tab.resize(self.width(),self.height())
        self.tab1 = QWidget()
        self.tab2 = QWidget()
        self.tab.addTab(self.tab1, 'Tab-1')
        self.tab.addTab(self.tab2, 'Tab-2')
        self.show()
if __name__ == '__main__':
    app = QApplication([])
    gui = MainWindow()
    app.exec_()

```

```

[1]: # kitabxanaları proqrama daxil edirik
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import sys

class MainWindow(QWidget):

    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        # sas pəncərə self ifadəsində referans alındı.
        self.setWindowTitle("Bu mənim ilk proqramım")
        # Pəncərə loqosu
        self.setWindowIcon(QIcon('../image/ico.png')) # Pəncərə sol yuxarı
        ↪kündə logo əlavə olundu
        # Pəncərə arxa plan rəngi
        self.palette = QPalette()
        self.palette.setColor(QPalette.Background, Qt.white)
        self.setPalette(self.palette)
        self.setGeometry(400,400,600,600)
        self.setFixedSize(600,600)
        self.tab = QTabWidget(self)
        self.tab.setGeometry(0,30,500,450)
        self.tab1 = QWidget()
        self.tab2 = QWidget()
        self.tab.addTab(self.tab1, 'Tab-1')
        self.tab.addTab(self.tab2, 'Tab-2')
        self.toolbar = QToolBar(self)
        self.toolbar.addAction('File')
        self.toolbar.addAction('Open')
        self.toolbar.setMovable(False)
        self.show()
if __name__ == '__main__':
    app = QApplication([])

```

```
gui = MainWindow()
app.exec_()
```

```
[ ]:
```

```
[2]: #Brauzerin hazırlanması
```

pip install PyQtWebEngine

```
[1]: from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtWebEngineWidgets import *
import sys,os

class Dialog(QDialog):

    def __init__(self, *args, **kwargs):
        super(Dialog, self).__init__(*args, **kwargs)

        QBtn = QDialogButtonBox.Ok
        self.buttonBox = QDialogButtonBox(QBtn)
        self.buttonBox.accepted.connect(self.accept)
        self.buttonBox.rejected.connect(self.reject)
        self.setWindowTitle('About')
        layout = QVBoxLayout()

        title = QLabel("QBrowser")
        font = title.font()
        font.setPointSize(20)
        title.setFont(font)

        layout.addWidget(title)

        logo = QLabel()
        logo.setPixmap(QPixmap('image/qbrowser.png'))
        layout.addWidget(logo)

        layout.addWidget(QLabel("Version 0.1"))
        layout.addWidget(QLabel("Copyright 2020 QBrowser Inc."))

        for i in range(0, layout.count()):
            layout.itemAt(i).setAlignment(Qt.AlignHCenter)

        layout.addWidget(self.buttonBox)
```

```

        self.setLayout(layout)

class window(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(window, self).__init__(*args, **kwargs)
        self.tabs = QTabWidget()
        self.tabs.setDocumentMode(True)
        self.tabs.tabBarDoubleClicked.connect(self.tab_open_doubleclick)
        self.tabs.currentChanged.connect(self.current_tab_changed)
        self.tabs.setTabsClosable(True)
        self.tabs.tabCloseRequested.connect(self.close_current_tab)
        self.setCentralWidget(self.tabs)

        navtb = QToolBar("Navigation")
        navtb.setIconSize(QSize(16, 16))
        self.addToolBar(navtb)

        back_btn = QAction(QIcon('image/back.png'), "Back", self)
        back_btn.setStatusTip("Back to previous page")
        back_btn.triggered.connect(lambda: self.tabs.currentWidget().back())
        navtb.addAction(back_btn)

        next_btn = QAction(QIcon('image/forward.png'), "Forward", self)
        next_btn.setStatusTip("Forward to next page")
        next_btn.triggered.connect(lambda: self.tabs.currentWidget().forward())
        navtb.addAction(next_btn)

        reload_btn = QAction(QIcon('image/reload.png'), "Reload", self)
        reload_btn.setStatusTip("Reload page")
        reload_btn.triggered.connect(lambda: self.tabs.currentWidget().reload())
        navtb.addAction(reload_btn)

        home_btn = QAction(QIcon('image/home.png'), "Home", self)
        home_btn.setStatusTip("Go home")
        home_btn.triggered.connect(self.navigate_home)
        navtb.addAction(home_btn)

        navtb.addSeparator()

        self.urlbar = QLineEdit()
        self.urlbar.returnPressed.connect(self.navigate_to_url)
        navtb.addWidget(self.urlbar)

        stop_btn = QAction(QIcon('image/stop.png'), "Stop", self)
        stop_btn.setStatusTip("Stop loading current page")
        stop_btn.triggered.connect(lambda: self.tabs.currentWidget().stop())
        navtb.addAction(stop_btn)

        self.menuBar().setNativeMenuBar(False)
        self.statusBar()

        file_menu = self.menuBar().addMenu("&File")

        new_tab_action = QAction(QIcon('image/newtab.jpg'), "New Tab", self)

```

```

new_tab_action.setStatusTip("Open a new tab")
new_tab_action.triggered.connect(lambda _: self.add_new_tab())
file_menu.addAction(new_tab_action)
open_file_action = QAction(QIcon('image/openfile.png'), "Open file...",
↪self)
open_file_action.setStatusTip("Open from file")
open_file_action.triggered.connect(self.open_file)
file_menu.addAction(open_file_action)
save_file_action = QAction(QIcon('image/saveas.png'), "Save Page As...",
↪, self)
save_file_action.setStatusTip("Save current page to file")
save_file_action.triggered.connect(self.save_file)
file_menu.addAction(save_file_action)
print_action = QAction(QIcon('image/print.png'), "Print...", self)
print_action.setStatusTip("Print current page")
print_action.triggered.connect(self.print_page)
file_menu.addAction(print_action)
help_menu = self.menuBar().addMenu("&Help")
about_action = QAction(QIcon('image/question.png'), "About QBrowser",
↪self)
about_action.setStatusTip("Find out more about QBrowser")
about_action.triggered.connect(self.about)
help_menu.addAction(about_action)
navigate_action = QAction(QIcon('image/qbrowser.png'), "QBrowser", self)
navigate_action.setStatusTip("Go to QBrowser Homepage")
navigate_action.triggered.connect(self.navigate_qbrowser)
help_menu.addAction(navigate_action)
self.add_new_tab(QUrl('http://www.google.com'), 'Homepage')
self.setWindowTitle("QBrowser")
self.setWindowIcon(QIcon('image/qbrowser.png'))
self.show()
def add_new_tab(self, qurl=None, label="Blank"):
    if qurl is None:
        qurl = QUrl('')
    browser = QWebEngineView()
    browser.setUrl(qurl)
    i = self.tabs.addTab(browser, label)
    self.tabs.setCurrentIndex(i)
    browser.urlChanged.connect(lambda qurl, browser=browser:
                                self.update_urlbar(qurl, browser))
    browser.loadFinished.connect(lambda _, i=i, browser=browser:
                                self.tabs.setTabText(i, browser.page().
↪title()))

def tab_open_doubleclick(self, i):
    if i == -1:

```



```

        self.add_new_tab()

def current_tab_changed(self, i):
    qurl = self.tabs.currentWidget().url()
    self.update_urlbar(qurl, self.tabs.currentWidget())
    self.update_title(self.tabs.currentWidget())

def close_current_tab(self, i):
    if self.tabs.count() < 2:
        return
    self.tabs.removeTab(i)

def update_title(self, browser):
    if browser != self.tabs.currentWidget():
        return
    title = self.tabs.currentWidget().page().title()
    self.setWindowTitle("{} - QBrowser".format(title))

def navigate_qbrowser(self):
    self.tabs.currentWidget().setUrl(QUrl("https://github.com/RashadGarayev/
↪QBrowser"))

def about(self):
    dialog = Dialog()
    dialog.exec_()

def open_file(self):
    filename, _ = QFileDialog.getOpenFileName(self, "Open file", "",
                                                "Hypertext Markup Language (*.
↪htm *.html);;"
                                                "All files (*.*)")

    if filename:
        with open(filename, 'r') as f:
            html = f.read()
        self.tabs.currentWidget().setHtml(html)
        self.urlbar.setText(filename)

def save_file(self):
    filename, _ = QFileDialog.getSaveFileName(self, "Save Page As", "",
                                                "Hypertext Markup Language (*.
↪htm *.html);;"
                                                "All files (*.*)")

    if filename:
        html = self.tabs.currentWidget().page().toHtml()
        with open(filename, 'w') as f:
            f.write(html.encode('utf8'))

```

```

def print_page(self):
    dialog = QPrintPreviewDialog()
    dialog.paintRequested.connect(self.browser.print_)
    dialog.exec_()

def navigate_home(self):
    self.tabs.currentWidget().setUrl(QUrl("http://www.google.com"))

def navigate_to_url(self):
    q = QUrl(self.urlbar.text())
    if q.scheme() == "":
        q.setScheme("http")

    self.tabs.currentWidget().setUrl(q)

def update_urlbar(self, q, browser=None):

    if browser != self.tabs.currentWidget():
        return
    self.urlbar.setText(q.toString())
    self.urlbar.setCursorPosition(0)

if __name__ == '__main__':
    app = QApplication([])
    gui = window()
    app.exec_()

```

[2]: *#Webcam*

```

[1]: from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtMultimedia import *
from PyQt5.QtMultimediaWidgets import *
import os
import sys
class MainWindow(QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.online_webcams = QCameraInfo.availableCameras()
        if not self.online_webcams:
            pass
        self.exist = QCameraViewfinder()
        self.exist.show()
        self.setCentralWidget(self.exist)

```

```

        self.get_webcam(0)
        self.setWindowTitle("WebCam")
        self.show()
    def get_webcam(self, i):
        self.my_webcam = QCamera(self.online_webcams[i])
        self.my_webcam.setViewfinder(self.exist)
        self.my_webcam.setCaptureMode(QCamera.CaptureStillImage)
        self.my_webcam.error.connect(lambda: self.alert(self.my_webcam.
↪errorString()))
        self.my_webcam.start()

    def alert(self, s):
        err = QErrorMessage(self)
        err.showMessage(s)
if __name__ == '__main__':

    app = QApplication(sys.argv)
    app.setApplicationName("WebCam")

    window = MainWindow()
    app.exec_()

```

```

[1]: import sys
from PyQt5.QtCore import QTimer
from PyQt5.QtWidgets import QApplication, QMainWindow, QProgressBar
from PyQt5.QtCore import Qt

class Example(QMainWindow):

    def __init__(self):
        super().__init__()

        self.pbar = QProgressBar(self)
        self.pbar.setGeometry(30, 40, 200, 25)
        self.pbar.setValue(50)

        self.setWindowTitle("QT Progressbar Example")
        self.setGeometry(32,32,320,200)
        self.show()

        self.timer = QTimer()
        self.timer.timeout.connect(self.handleTimer)
        self.timer.start(1000)

    def handleTimer(self):

```

```

        value = self.pbar.value()
        if value < 100:
            value = value + 1
            self.pbar.setValue(value)
        else:
            self.timer.stop()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = Example()
    sys.exit(app.exec_())

```

An exception has occurred, use %tb to see the full traceback.

**SystemExit: 0**

C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3339: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
 warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

[ ]: