

សំនាំដើម

កញ្ចប់ PyQt4

PyQt4 គឺជាកញ្ចប់មួយដែលនៅក្នុងនោះមានក្បួនខ្នាតជាច្រើនដែលអនុញ្ញាតឲ្យយើងអាចសរសេរកម្មវិធីភាសា Python បង្កើតជារូបបង្អួចផ្សេងៗដែលមានមូលដ្ឋានគ្រឹះជាប្រព័ន្ធបង្អួច Qt ។ ដូចនេះ PyQt4 គឺជា **សម្ភារៈ** (toolkit) មួយចំនួនប្រើសម្រាប់សរសេរកម្មវិធីភាសា Python បង្កើត **កម្មវិធីជាក់ស្តែង** (application) ទាំងឡាយណាដែលមានលក្ខណៈជារូបបង្អួច។

នៅក្នុងប្រព័ន្ធប្រតិបត្តិការ Windows Vista ដើម្បីអាចសរសេរកម្មវិធីបង្កើតប្រព័ន្ធបង្អួចដោយប្រើសម្ភារៈ PyQt4 បាន យើងត្រូវមានផ្នែកទន់ Python 3 និងកញ្ចប់ PyQt4 នៅក្នុងកំពូទ័ររបស់យើង។ យើងអាចទាញយកផ្នែកទន់ Python 3 ពីគេហទំព័រ www.python.org និងកញ្ចប់ PyQt4 ពីគេហទំព័រ <http://www.riverbankcomputing.co.uk/software/pyqt/download> មកតម្លើងនៅក្នុងកំពូទ័ររបស់យើង។

លុះមានផ្នែកទន់ទាំងនេះហើយ យើងអាចប្រើកម្មវិធី IDLE ដែលមាននៅក្នុងផ្នែកទន់ Python3 ជារោងជាងសម្រាប់សរសេរកម្មវិធីភាសា Python ។ ក៏ប៉ុន្តែគួរឲ្យស្តាយដែរ កម្មវិធី IDLE នេះមិនអនុញ្ញាតឲ្យសរសេរអក្សរខ្មែរយូនីកូដដោយផ្ទាល់នៅលើនោះបានទេ គឺយើងត្រូវសរសេរអក្សរខ្មែរយូនីកូដដោយប្រើកម្មវិធីផ្សេងទៀត រួចសឹមចម្លងយកមកដាក់នៅក្នុងកម្មវិធី IDLE នេះ។ ដើម្បីសរសេរអក្សរខ្មែរយូនីកូដ យើងអាចប្រើកម្មវិធីជាច្រើនមានដូចជា Notepad ឬ OpenOffice ជាដើម។

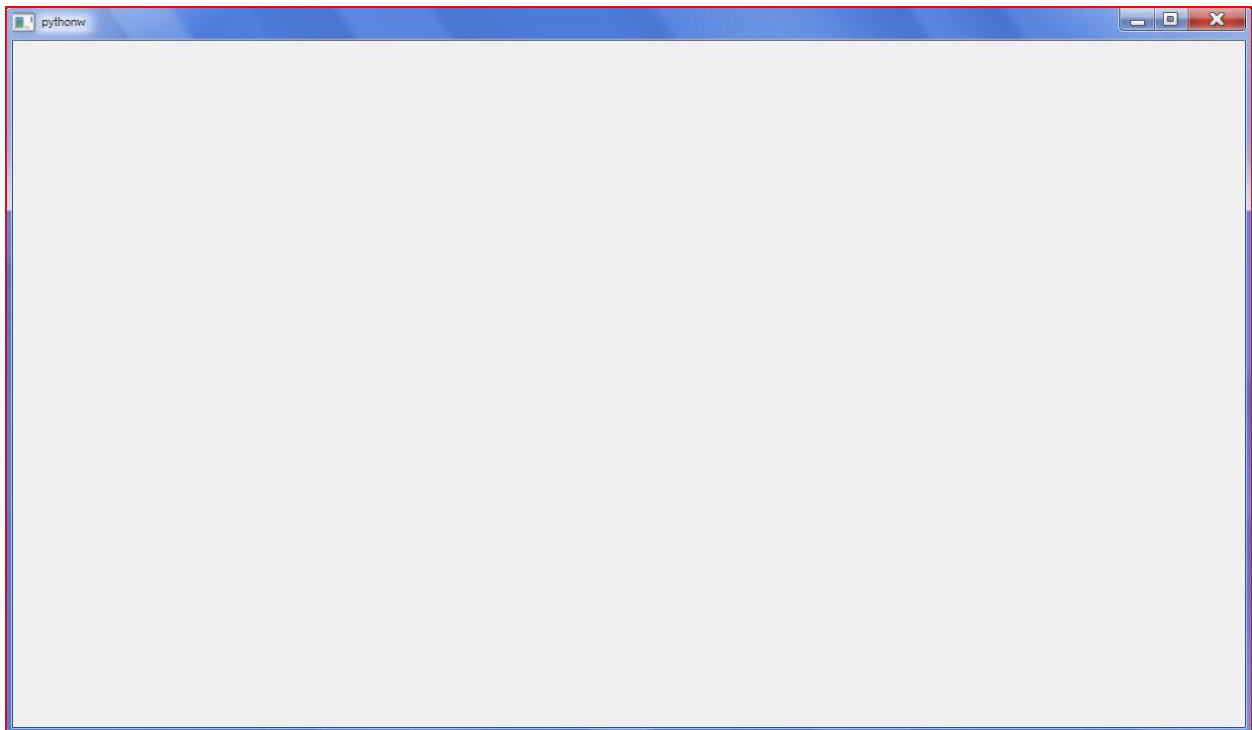
កម្មវិធី PyQt4 ជាអប្បបរិមា

ជាជំហានដំបូង យើងអាចចាប់ផ្តើមសរសេរកម្មវិធី PyQt4 ជាអប្បបរិមាដូចខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

កម្មវិធី = QApplication(sys.argv)

បង្កើត = QWidget()
បង្កើត.show()
កម្មវិធី.exec_()
```



ខាងលើនេះគឺជាកម្មវិធី PyQt4 ជាអប្បបរមាដែលនៅក្នុងនោះមានបញ្ហាចាំបាច់មួយចំនួន ដែលជាមូលដ្ឋានគ្រឹះនៃគ្រប់កម្មវិធី PyQt4 ទាំងអស់។ បញ្ហាទាំងនោះមានតួនាទីដូចខាង ក្រោមនេះ៖

`import sys` គឺជាការយកសាស្ត្រាឈ្មោះ `sys` នៅក្នុងបណ្ណាល័យមជ្ឈឹមមកប្រើ។

`from PyQt4.QtCore import*` គឺជាបញ្ជាតម្រូវឲ្យចម្លងយកអ្វីៗទាំងអស់ដែលមាននៅក្នុងសាស្ត្រា QtCore នៃកញ្ចប់ PyQt4 មកទុកយកទៅប្រើនៅពេលក្រោយ។

`from PyQt4.QtGui import*` គឺជាបញ្ជាតម្រូវឲ្យចម្លងយកអ្វីៗទាំងអស់ដែលមាននៅក្នុងសាស្ត្រា QtGui នៃកញ្ចប់ PyQt4 មកទុកយកទៅប្រើនៅពេលក្រោយ។

`កម្មវិធី = QApplication(sys.argv)` គឺជាការបង្កើតវត្ថុឈ្មោះ កម្មវិធី មួយដែលជាសិស្សនៃថ្នាក់ QApplication ។ ដំណឹងសម្រាប់ដំណាងនៅក្នុងស្ថាបនិកនៃថ្នាក់ QApplication គឺជាសម្បត្តិ ឈ្មោះ `argv` នៅក្នុងសាស្ត្រាឈ្មោះ `sys` ។ យើងនឹងបានស្គាល់ពីសម្បត្តិឈ្មោះ `argv` នៅពេល ខាងមុខនេះ។

`បង្អួច = QWidget()` គឺជាការបង្កើតរូបបង្អួចមួយមានឈ្មោះថា បង្អួច ដោយយកថ្នាក់ឈ្មោះ QWidget មកប្រើ។

`បង្អួច.show()` គឺជាការយកវិធីឈ្មោះ `show` មកប្រើដើម្បីធ្វើឲ្យរូបបង្អួចឈ្មោះ បង្អួច ដែលទើប នឹងត្រូវបានបង្កើតនោះលេចចេញមានរូបរាងឡើង។ ពីព្រោះក្រោយពីត្រូវបានបង្កើតរួចហើយ រូបបង្អួចមានតែនៅក្នុងសតិរបស់កំពូទ័រតែប៉ុណ្ណោះ គឺយើងមិនអាចមើលវាឃើញឡើយ។ ហើយបើចង់ឲ្យបង្អួចនោះលេចចេញរូបរាងឡើង យើងត្រូវប្រើវិធីឈ្មោះ `show` នេះ។

កម្មវិធី.exec_() គឺជាការយកវិធីឈ្មោះ exec_ មកប្រើដើម្បីធ្វើឲ្យកម្មវិធី PyQt4 មានដំណើរការ ជាប់មិនដាច់រហូតដល់មានការចុចបិទបង្អួចដែលលេចឡើងនោះ។

សរុបមក ក្នុងគ្រប់កម្មវិធី PyQt4 ទាំងអស់ទាមទារឲ្យមានវត្ថុពីរយ៉ាងគឺ៖ វត្ថុដែលជាសិស្សនៃ ថ្នាក់ QApplication ឬថ្នាក់របស់វា និងវត្ថុដែលជាសិស្សនៃថ្នាក់ QWidget ឬថ្នាក់របស់វា ។ វត្ថុដែលជាសិស្សនៃថ្នាក់ QApplication មាននាទីផ្តល់លទ្ធភាពឲ្យយើងអាចវិធីឈ្មោះ exec_ មកប្រើដើម្បីធានាឲ្យកម្មវិធី PyQt4 មានដំណើរទៅជាប់មិនដាច់។ ចំណែកវត្ថុដែលជា សិស្សនៃថ្នាក់ QWidget គឺជារូបបង្អួចដែលដំណាច់ឲ្យកម្មវិធី PyQt4 ទាំងមូលតែម្តង។

ប្រព័ន្ធបង្អួច

បើយើងពិនិត្យមើលកម្មវិធីជាក់ស្តែងមួយចំនួនមានដូចជា open office ជាដើម យើងនឹងឃើញថា កម្មវិធីទាំងនោះមានមូលដ្ឋានគ្រឹះជាបង្អួចធំមួយដែលនៅក្នុងនោះមាន *វិប* (widget) តូចៗមួយចំនួនទៀត មានដូចជា៖ menu, button, toolbar ជាដើម។ល។

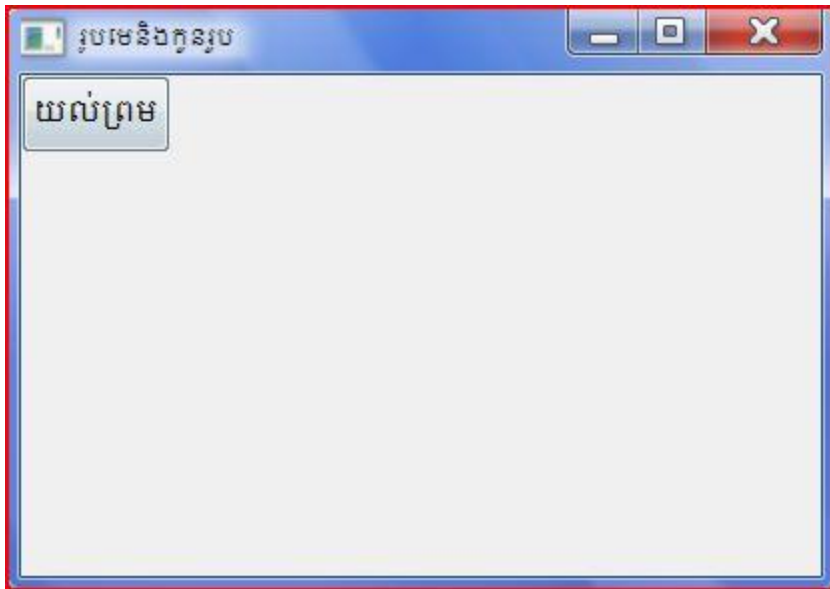
បង្អួចធំនិងរូបតូចៗនៅលើនោះមានអន្តរទំនាក់ទំនងជាមួយគ្នាយ៉ាងជិតស្និទ្ធ ពីព្រោះបើនៅពេលណាដែលរូបណាមួយទទួលរងនូវអំពើអ្វីមួយ រូបផ្សេងៗទៀតក៏ទទួលរងនូវផលប៉ះពាល់ណាមួយដែរ។ ដូចនេះ រូបទាំងនោះរួមផ្សំគ្នាបង្កើតជាប្រព័ន្ធមួយដែលយើងអាចឲ្យឈ្មោះថា *ប្រព័ន្ធបង្អួច* (window system) ។ យ៉ាងណាមិញពាក្យថា window system នេះត្រូវបានយកទៅប្រើប្រាស់ទន្ទឹមគ្នានឹងពាក្យថា Graphic User Interface នៅក្នុងផ្នែកទំនាក់ទំនងមួយចំនួន មានដូចជា Blender ជាដើម។ ដូចនេះការប្រើប្រាស់ពាក្យថាប្រព័ន្ធបង្អួចគឺសំដៅទៅលើពាក្យអង់គ្លេស window system ឬ Graphic User Interface នេះតែម្តង។

ផ្ទាំងរូបនិងកូនរូប

នៅក្នុងកម្មវិធីដែលមានលក្ខណៈជាប្រព័ន្ធបង្អួច រូបធំដែលនៅក្នុងនោះមានរូបតូចៗជាច្រើនទៀតត្រូវហៅថា *វិបមេ* (parent widget) ចំណែកឯរូបតូចៗនៅក្នុងនោះវិញត្រូវហៅថា *កូនរូប* (child widget) ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*
```

```
កម្មវិធី = QApplication(sys.argv)
បង្កើត = QWidget()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.resize(400, 250)
បង្កើត.setWindowTitle("រូបមេនិងកូនរូប")
គន្លឹះ = QPushButton("យល់ព្រម", បង្កើត)
បង្កើត.show()
កម្មវិធី.exec_()
```



`បង្កើត = QWidget()` គឺជាការយកថ្នាក់ឈ្មោះ `QWidget` មកប្រើដើម្បីបង្កើតរូបបង្អួចមួយដែល ជារូបមេ ព្រោះនៅក្នុងនោះមានរូបតូចមួយទៀត។

កម្មវិធី.setFont(QFont('Khmer OS', 11)) គឺជាការយកវិធីឈ្មោះ: setFont មកប្រើដើម្បីកំណត់ យកពុម្ពអក្សរខ្មែរយូនីកូដឈ្មោះ: Khmer OS ទំហំ 11 ជាពុម្ពអក្សរដែលត្រូវយកទៅប្រើនៅក្នុង កម្មវិធីទាំងមូល។

បង្អួច.resize(400, 250) គឺជាការយកវិធីឈ្មោះ: resize មកប្រើដើម្បីកំណត់ឲ្យបង្អួចឈ្មោះ: បង្អួច នៅក្នុងកម្មវិធីខាងលើមានទំហំ (400, 250) ។

បង្អួច.setWindowTitle("រូបមេនិងកូនរូប") គឺជាការយកវិធីឈ្មោះ: setWindowTitle មកប្រើ ដើម្បីកំណត់យកកម្រងអក្សរ "រូបមេនិងកូនរូប" ជាចំណងជើងដែលត្រូវសរសេរនៅលើរបារ ចំណងជើងរបស់បង្អួចឈ្មោះ: បង្អួច ។

គន្លឹះ = QPushButton("យល់ព្រម", បង្អួច) គឺជាការយកថ្នាក់ឈ្មោះ: QPushButton មកប្រើ ដើម្បីបង្កើតកូនរូបមួយដែលជាគន្លឹះ: (pushbutton) មានឈ្មោះថា គន្លឹះ ។ គន្លឹះនោះត្រូវបាន យកទៅបិទភ្ជាប់ទៅនឹងរូបមេដែលជាបង្អួចមានឈ្មោះថា បង្អួច ។ នៅលើគន្លឹះនោះមានពាក្យ ថា "យល់ព្រម" ដែលជាកម្រងអក្សរត្រូវបានផ្តល់ឲ្យទៅស្ថាបនិកនៃថ្នាក់ឈ្មោះ: QPushButton នៅពេលដែលថ្នាក់នោះត្រូវបានយកមកប្រើ។

ដូចនេះ យើងសង្កេតឃើញថា នៅក្នុងកម្មវិធីខាងលើនេះ មានការបង្កើតរូបពីរគឺរូបឈ្មោះ: បង្អួច ដែលមានលក្ខណៈជារូបមេ និងរូបឈ្មោះ: គន្លឹះ ដែលមានលក្ខណៈជាកូនរូប ព្រោះវា ត្រូវបានយកទៅបិទនៅក្នុងរូបមេមានឈ្មោះថា បង្អួច នោះ។

ការបង្កើត ថ្នាក់នៃថ្នាក់ QWidget

ការបង្កើតរូបមេដែលជារូបបង្អួចនៅក្នុងកម្មវិធីខាងលើ ប្រព្រឹត្តទៅដោយមានការយកថ្នាក់
ឈ្មោះ QWidget មកប្រើដោយផ្ទាល់តែម្តង។ តែជាទូទៅ គេច្រើនបង្កើតរូបបង្អួចដោយយក
ថ្នាក់រងនៃថ្នាក់ឈ្មោះ QWidget នេះមកប្រើ។ បានន័យថា ជាដំបូងយើងត្រូវបង្កើតថ្នាក់រងនៃ
ថ្នាក់ឈ្មោះ QWidget ជាមុនសិន រួចបានយកថ្នាក់រងនោះមកប្រើជាក្រោយដើម្បីបង្កើតសិស្ស
ដែលជារូបបង្អួចណាមួយនោះ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.resize(400, 250)
        សិស្ស.setWindowTitle("រូបមេនិងកូនរូប")
        គន្ធី៖ = QPushButton("យល់ព្រម", សិស្ស)
        បង្អួច.resize(400, 250)

    កម្មវិធី = QApplication(sys.argv)
    បង្អួច = រូបមេ()
    កម្មវិធី.setFont(QFont('Khmer OS', 11))
    បង្អួច.show()
    កម្មវិធី.exec_()
```

class រូបមេ(QWidget): គឺជាបង្កើតថ្នាក់ឈ្មោះ រូបមេ មួយដែលជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ
QWidget ។

បង្អួច = រូបមេ() គឺជាការយកថ្នាក់រងឈ្មោះ រូបមេ មកប្រើដើម្បីបង្កើតរូបបង្អួចមួយមានឈ្មោះ ថា បង្អួច ។ ប្រការនេះធ្វើឲ្យស្ថាបនិកនៃថ្នាក់រងនោះត្រូវយកមកប្រើដែលជាហេតុបណ្តាលឲ្យ បញ្ហាផ្សេងៗនៅក្នុងស្ថាបនិកនោះក៏ត្រូវយកមកអនុវត្តដែរ។ ប្រការនេះធ្វើឲ្យគន្លឹះឈ្មោះ គន្លឹះ ក៏ត្រូវបានបង្កើតឡើងនៅពេលនោះដែរ។

នៅក្នុងកម្មវិធីខាងលើនេះ យើងសង្កេតឃើញថា ការកំណត់ពីទំហំនិងចំណងជើងរបស់បង្អួច និងការបង្កើតគន្លឹះ គឺត្រូវបានធ្វើឡើងនៅក្នុងស្ថាបនិកនៃថ្នាក់រងឈ្មោះ រូបមេ នោះទាំងអស់។ ហើយឈ្មោះ សិស្ស នៅក្នុងស្ថាបនិកនៃថ្នាក់រងនោះ គឺជាឈ្មោះមួយទៀតរបស់បង្អួចឈ្មោះ បង្អួច ព្រោះបង្អួចនេះត្រូវបានផ្តល់ឲ្យទៅស្ថាបនិកសម្រាប់ដំណាង សិស្ស នោះជាស្វ័យប្រវត្តិ ។

ក្រឡាទំហំ

ក្រឡាទំហំ (layout) គឺជារូបមើលមិនឃើញម៉្យាងប្រើសម្រាប់យករូបផ្សេងៗទៀតទៅតម្រៀប នៅក្នុងនោះមុននឹងយកទៅបិទនៅលើរូបមេណាមួយ។ ក្រឡាទំហំចែកចេញជាច្រើនប្រភេទ ដូចខាងតទៅនេះ៖

ប្រអប់ទំហំ

ប្រអប់ទំហំ (box layout) គឺជាក្រឡាទំហំម្យ៉ាងប្រើសម្រាប់តម្រៀបរូបផ្សេងៗតាមខ្សែដេកឬ ខ្សែឈរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*
```

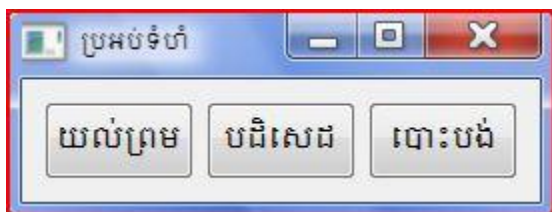
```

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
        គន្លឹះ = []
        for ផ្លាក in ["យល់ព្រម", "បដិសេធ", "បោះបង់"]:
            គន្លឹះ.append(QPushButton(ផ្លាក))

        ប្រអប់ទំហំដេក = QHBoxLayout()
        for លេខរៀង in range(3):
            ប្រអប់ទំហំដេក.addWidget(គន្លឹះ[លេខរៀង])
        សិស្ស.setLayout(ប្រអប់ទំហំដេក)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



ប្រអប់ទំហំដេក = QHBoxLayout() គឺជាការយកថ្នាក់ឈ្មោះ QHBoxLayout មកប្រើដើម្បីបង្កើតប្រអប់ទំហំសម្រាប់តម្រៀបរូបតាមខ្សែដេកមួយមានឈ្មោះថា ប្រអប់ទំហំដេក ។

`ប្រអប់ទំហំដេក.addWidget(គន្លឹះ[លេខរៀង])` គឺជាការយកវិធីឈ្មោះ `addWidget` មកប្រើដើម្បី តម្រៀបគន្លឹះចំនួនបីតាមខ្សែដេកនៅក្នុងប្រអប់ទំហំឈ្មោះ ប្រអប់ទំហំដេក ។

`សិស្ស.setLayout(ប្រអប់ទំហំដេក)` គឺជាការយកវិធីឈ្មោះ `setLayout` មកប្រើដើម្បីបិទប្រអប់ ទំហំឈ្មោះ ប្រអប់ទំហំដេក ភ្ជាប់ទៅនឹងបង្អួចឈ្មោះ សិស្ស ។

ក្រៅពីការតម្រៀបរូបតាមខ្សែដេក យើងក៏អាចយករូបផ្សេងៗមកតម្រៀបនៅក្នុងប្រអប់ទំហំ តាមខ្សែឈរបានដែរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
        គន្លឹះ = [ ]
        for ផ្លាក in ["យល់ព្រម", "បដិសេធ", "បោះបង់"]:
            គន្លឹះ.append(QPushButton(ផ្លាក))

        ប្រអប់ទំហំឈរ = QVBoxLayout()
        for លេខរៀង in range(3):
            ប្រអប់ទំហំឈរ.addWidget(គន្លឹះ[លេខរៀង])
        សិស្ស.setLayout(ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)

```

```

បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



ប្រអប់ទំហំឈរ = QVBoxLayout() គឺជាការបង្កើតប្រអប់ទំហំសម្រាប់តម្រៀបរូបតាមខ្សែឈរ មួយមានឈ្មោះថា ប្រអប់ទំហំឈរ ។

រូបដែលយើងយកទៅតម្រៀបនៅក្នុងប្រអប់ទំហំ អាចជារូបប្រភេទណាក៏បានដែរ។ ហើយ ដើម្បីអាចតម្រៀបរូបតាមខ្សែដេកផង ខ្សែឈរផង គេច្រើនយកប្រអប់ទំហំមួយទៅតម្រៀបជា រូបនៅក្នុងប្រអប់ទំហំមួយទៀតដោយធ្វើដូចខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)

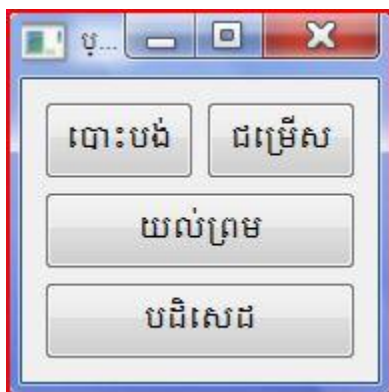
```

```

សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
គន្លឹះ = [ ]
for ផ្លាក in ["យល់ព្រម", "បដិសេធ", "បោះបង់", "ជម្រើស"]:
    គន្លឹះ.append(QPushButton(ផ្លាក))
ប្រអប់ទំហំឈរ = QVBoxLayout()
ប្រអប់ទំហំដេក = QHBoxLayout()
for លេខរៀង in range(4):
    if លេខរៀង < 2:
        ប្រអប់ទំហំឈរ.addWidget(គន្លឹះ[លេខរៀង])
    else:
        ប្រអប់ទំហំដេក.addWidget(គន្លឹះ[លេខរៀង])
ប្រអប់ទំហំឈរ.insertLayout(0, ប្រអប់ទំហំដេក)
សិស្ស.setLayout(ប្រអប់ទំហំឈរ)
    
```

```

កម្មវិធី = QApplication(sys.argv)
បង្អួច = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្អួច.show()
កម្មវិធី.exec_()
    
```



ប្រអប់ទំហំឈរ.insertLayout(0, ប្រអប់ទំហំដេក) គឺជាការយកវិធីឈ្មោះ insertLayout មកប្រើ ដើម្បីបញ្ចូលប្រអប់ទំហំឈ្មោះ ប្រអប់ទំហំដេក ទៅក្នុងប្រអប់ទំហំឈ្មោះ ប្រអប់ទំហំឈរ នៅ ត្រង់ចំនុចដែលមានលេខរៀង 0 ។

ស្ថាបនិកនៃថ្នាក់ QVBoxLayout

ស្ថាបនិកនៃថ្នាក់ឈ្មោះ QVBoxLayout មានទម្រង់ដូចខាងក្រោមនេះ៖

- `__init__(self)`
- `__init__(self, QWidget parent)`

ស្ថាបនិកនៃថ្នាក់ QHBoxLayout

- `__init__(self)`
- `__init__(self, QWidget parent)`

វិធីនៃថ្នាក់ QBoxLayout

ថ្នាក់ឈ្មោះ QVBoxLayout និងថ្នាក់ឈ្មោះ QHBoxLayout សុទ្ធតែជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ QBoxLayout ។ ដូចនេះតាមរយៈសិស្សនៃថ្នាក់ឈ្មោះ QVBoxLayout ឬសិស្សនៃថ្នាក់ឈ្មោះ QHBoxLayout យើងអាចយកវិធីនៅក្នុងថ្នាក់ឈ្មោះ QBoxLayout មកប្រើបានតាមសេចក្តី ត្រូវការ។ វិធីសំខាន់ៗមាននៅក្នុងថ្នាក់ឈ្មោះ QBoxLayout មានដូចខាងក្រោមនេះ៖

- `QBoxLayout.addWidget (self, QWidget, int stretch = 0, Qt.Alignment alignment = 0)` : គឺ ជាវិធីប្រើសម្រាប់យករូបណាមួយទៅតម្រៀបនៅខាងចុងគេនៅក្នុងប្រអប់ទំហំរូបណា មួយ។

- **stretch** : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាកត្តាធ្វើឲ្យរូបត្រូវយកទៅតម្រៀបយឺតលូតវែងទៅតាមទិសដៅនៃការយឺតលូតវែងនៃប្រអប់ទំហំ នៅពេលដែលយើងចុចទាញពង្រីកបង្អួច។ លេខ 0 គឺជាកត្តាដែលធ្វើឲ្យរូបយឺតលូតវែងពេញផ្ទៃបង្អួច ហើយបើនៅក្នុងក្រឡាទំហំមានរូបលើសពីមួយ រូបទាំងអស់នោះនឹងរីកលូតស្មើគ្នា។ តែបើតម្លៃលេខសម្រាប់ដំណាង stretch ខុសពី 0 ហើយនៅក្នុងក្រឡាទំហំមានរូបជាច្រើននៅក្នុងនោះ រូបទាំងឡាយនឹងត្រូវរីកលូតអាស្រ័យទៅតាមប្រព័ន្ធសមមាត្រដោយធៀបទៅនឹងរូបផ្សេងទៀត។
- **alignment** : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាកត្តាកំណត់ពីទីតាំងរបស់រូបនៅក្នុងក្រឡារបស់វា។ ទីតាំងទាំងនោះមានដូចខាងក្រោមនេះ៖
 - **Qt.AlignLeft** : ទីតាំងជាប់នឹងជ្រុងខាងឆ្វេង
 - **Qt.AlignRight** : ទីតាំងជាប់នឹងជ្រុងខាងស្តាំ
 - **Qt.AlignCenter** : ទីតាំងជាប់នៅកណ្តាល
 - **Qt.AlignTop** : ទីតាំងខាងលើ
 - **Qt.AlignBotton** : ទីតាំងខាងក្រោម
 - **Qt.AlignHCenter** : ទីតាំងនៅកណ្តាលតាមខ្សែដេក
 - **Qt.AlignVCenter** : ទីតាំងនៅកណ្តាលតាមខ្សែឈរ
 - **Qt.AlignJustify** : ទីតាំងសមស្របណាមួយសម្រាប់កម្រងអក្សរ

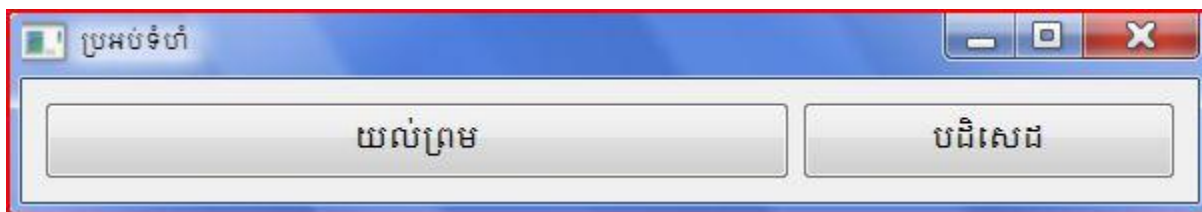
ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*
```

```

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
        សិស្ស.គន្លឹះ០ = QPushButton("យល់ព្រម")
        សិស្ស.គន្លឹះ១ = QPushButton("បដិសេធ")
        សិស្ស.ប្រអប់ទំហំដេក = QHBoxLayout()
        សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ០, 2)
        សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ១, 1)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំដេក)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
    
```



សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ០, 2) គឺជាការយកវិធីឈ្មោះ addWidget មក ប្រើដើម្បីយកគន្លឹះឈ្មោះ គន្លឹះ០ ទៅតម្រៀបនៅក្នុងប្រអប់ទំហំឈ្មោះ ប្រអប់ទំហំដេក ។

លេខ 2 ត្រូវបានផ្តល់ជាដំណឹងសម្រាប់ដំណាង stretch ។ កត្តានេះនឹងធ្វើឲ្យគន្លឹះនោះត្រូវរីកលូតអាស្រ័យទៅតាមប្រព័ន្ធសមមាត្រដោយធៀបទៅនឹងរូបផ្សេងទៀត។

សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ១, 1) គឺជាការយកវិធីឈ្មោះ `addWidget` មកប្រើដើម្បីយកគន្លឹះឈ្មោះ `គន្លឹះ១` ទៅតម្រៀបនៅក្នុងប្រអប់ទំហំឈ្មោះ `ប្រអប់ទំហំដេក` ។

លេខ 1 ត្រូវបានផ្តល់ជាដំណឹងសម្រាប់ដំណាង stretch ។ កត្តានេះនឹងធ្វើឲ្យគន្លឹះនោះត្រូវរីកលូតអាស្រ័យទៅតាមប្រព័ន្ធសមមាត្រដោយធៀបទៅនឹងរូបផ្សេងទៀត។

នៅពេលដែលកម្មវិធីខាងលើនេះកំពុងតែដំណើរការ បើយើងចុចជាយបង្អួចទាញពង្រីក គន្លឹះឈ្មោះ `គន្លឹះ០` នឹងរីកលូតទៅតាមផលធៀប $2/3$ ចំណែកឯគន្លឹះចុចឈ្មោះ `គន្លឹះ១` វិញនឹងរីកលូតទៅតាមផលធៀប $1/3$ ។

- `QBoxLayout.addSpacing (self, int size)` : គឺជាវិធីប្រើសម្រាប់ធ្វើឲ្យមានគំលាតរវាងរូបផ្សេងៗ។
 - `size` : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាកត្តាកំណត់ពីគំលាតរវាងរូបទាំងឡាយ។

ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
        សិស្ស.គន្លឹះ០ = QPushButton("យល់ព្រម")
```

```

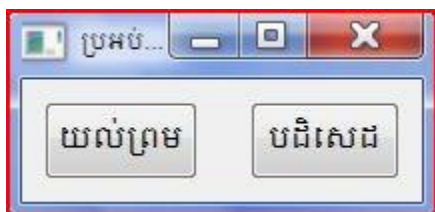
សិស្ស.គន្លឹះ១ = QPushButton("បដិសេធ")
សិស្ស.ប្រអប់ទំហំដេក = QHBoxLayout()
សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ០, 2)
សិស្ស.ប្រអប់ទំហំដេក.addWidget(សិស្ស.គន្លឹះ១, 1)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំដេក)

```

```

កម្មវិធី = QApplication(sys.argv)
បង្ហាញ = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្ហាញ.show()
កម្មវិធី.exec_()

```



`សិស្ស.ប្រអប់ទំហំដេក.addWidget(20)` គឺជាការយកវិធីឈ្មោះ `addWidget` មកប្រើដើម្បីធ្វើឲ្យមានគំលាតប្រវែង 20 ចំណុចរវាងគន្លឹះឈ្មោះ គន្លឹះ០ និងគន្លឹះឈ្មោះ គន្លឹះ១ ។

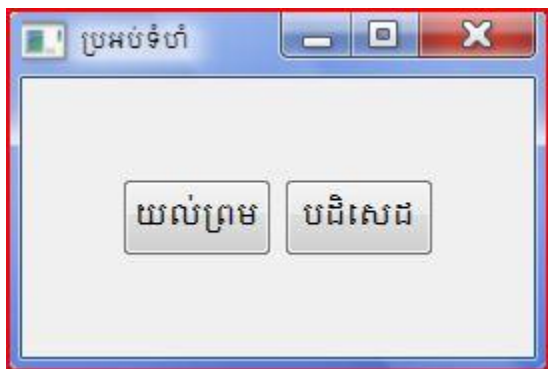
- `BoxLayout.addStretch (self, int stretch = 0)` : គឺជាវិធីប្រើសម្រាប់បន្ថែមគំលាតចូលក្នុងប្រអប់ទំហំ។ វិធីនេះស្រដៀងនឹងវិធីឈ្មោះ `addWidget` ដែរ តែវាខុសពីវិធីឈ្មោះ `addWidget` នៅត្រង់ថា គំលាតបានមកពីការប្រើវិធីឈ្មោះ `addStretch` អាចរីកលូតបាននៅពេលដែលយើងពង្រីកបង្ហាញ។

- `QBoxLayout.addWidget (self, QWidget widget, int stretch = 0)` : គឺជាវិធីប្រើសម្រាប់យកប្រអប់ទំហំណាមួយទៅតម្រៀបធ្វើជារូបនៅក្នុងប្រអប់ទំហំណាមួយផ្សេងទៀត។
- `QLayout.setContentsMargins (self, int left, int top, int right, int bottom)` : គឺជាវិធីប្រើសម្រាប់ធ្វើមានគំលាតនៅជុំវិញប្រអប់ទំហំទាំងមូល។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(self):
        QWidget.__init__(self)
        self.setWindowTitle("ប្រអប់ទំហំ")
        self.button0 = QPushButton("យល់ព្រម")
        self.button1 = QPushButton("បដិសេដ")
        self.layout = QHBoxLayout()
        self.layout.addWidget(self.button0, 2)
        self.layout.addWidget(self.button1, 1)
        self.setLayout(self.layout)

    កម្មវិធី = QApplication(sys.argv)
    បង្កើត = រូបមេ()
    កម្មវិធី.setFont(QFont('Khmer OS', 11))
    បង្កើត.show()
    កម្មវិធី.exec_()
```



សិស្ស.ប្រអប់ទំហំដេក.setContentsMargins(50, 50, 50, 50) គឺជាការយកវិធីឈ្មោះ:

setContentsMargins មកប្រើដើម្បីកំណត់ឲ្យមានគំលាតប្រវែង 50 ចំណុចនៅជុំវិញប្រអប់ទំហំឈ្មោះ ប្រអប់ទំហំដេក ដែលមានគន្លឹះចំនួនពីរនៅក្នុងនោះ។

- `QBoxLayout.spacing (self, int spacing)` : គឺជាវិធីប្រើសម្រាប់ធ្វើឲ្យមានគំលាតរវាងរូបទាំងអស់នៅក្នុងប្រអប់ទំហំណាមួយ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ប្រអប់ទំហំ")
        សិស្ស.គន្លឹះ០ = QPushButton("យល់ព្រម")
        សិស្ស.គន្លឹះ១ = QPushButton("បដិសេធ")
        សិស្ស.គន្លឹះ២ = QPushButton("បោះបង់")
        សិស្ស.ប្រអប់ទំហំដេក = QHBoxLayout()
```

```

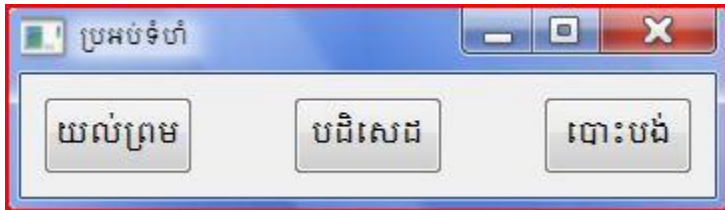
សិស្ស.ប្រអប់ទំហំដៃក.setSpacing(50)
សិស្ស.ប្រអប់ទំហំដៃក.addWidget(សិស្ស.គន្លឹះ០)
សិស្ស.ប្រអប់ទំហំដៃក.addWidget(សិស្ស.គន្លឹះ១)
សិស្ស.ប្រអប់ទំហំដៃក.addWidget(សិស្ស.គន្លឹះ២)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំដៃក)

```

```

កម្មវិធី = QApplication(sys.argv)
បង្ហាញ = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្ហាញ.show()
កម្មវិធី.exec_()

```



សិស្ស.ប្រអប់ទំហំដៃក.setSpacing(50) គឺជាការយកវិធីឈ្មោះ setSpacing មកប្រើដើម្បីធ្វើឲ្យ រូបទាំងអស់នៅក្នុងប្រអប់ទំហំឈ្មោះ QHBoxLayout ឃ្លាតពីគ្នាប្រវែង 50 ចំណុច។

- `QLayout.removeWidget (self, QWidget w)` : គឺជាវិធីប្រើសម្រាប់យករូបណាមួយចេញពី ក្នុងប្រអប់ទំហំ។
- `int QHBoxLayout.count (self)` : គឺជាវិធីប្រើសម្រាប់រាប់ចំនួនរូបនៅក្នុងប្រអប់ទំហំ។

- `Direction QBoxLayout.direction (self)` : គឺជាវិធីប្រើសម្រាប់ចម្លងយកទិសដៅនៃការរីកលូតរបស់ប្រអប់ទំហំ។ ទិសដៅទាំងនោះគឺជាចំនួនគត់ដែលជាសម្បត្តិមានឈ្មោះដូចខាងក្រោមនេះ៖
 - `QBoxLayout.LeftToRight = 0`
 - `QBoxLayout.RightToLeft = 1`
 - `QBoxLayout.TopToBottom = 2`
 - `QBoxLayout.BottomToTop = 3`
- `QBoxLayout.setDirection (self, Direction)` : គឺជាវិធីប្រើសម្រាប់កំណត់ទិសដៅនៃការរីកលូតរបស់ក្រឡាទំហំណាមួយ។

រលកសញ្ញា

ការតភ្ជាប់ រលកសញ្ញាទៅនឹង រន្ធដោត

នៅក្នុងកម្មវិធី PyQt4 បើសិនជាយើងចុចរូបណាមួយដែលមាននៅក្នុងប្រព័ន្ធបង្អួច រលកសញ្ញា (signal) ប្រភេទណាមួយនឹងត្រូវ ផ្សាយចេញ (emit) ពីរូបដែលត្រូវបានចុចនោះ។ ជាក់ស្តែង បើសិនជាយើងចុចរូបដែលជាគន្លឹះនៅក្នុងកម្មវិធី PyQt4 រលកសញ្ញាមួយប្រភេទ ដែលមានឈ្មោះថា "clicked()" នឹងត្រូវផ្សាយចេញពីរូបដែលជាគន្លឹះនោះ។ ហើយនៅពេល ដែលរលកសញ្ញាត្រូវផ្សាយចេញពីរូបផ្សេងៗ បើសិនជាយើងមិនអើពើទេ វានឹងគ្មានរឿងអ្វី កើតឡើងឡើយ រលកសញ្ញាទាំងនោះនឹងបាត់អស់ទៅវិញដោយឯកឯង។ ដើម្បីទាញយក ផលប្រយោជន៍ពីរលកសញ្ញាណាមួយ យើងត្រូវ តភ្ជាប់ (connect) រលកសញ្ញានោះទៅនឹង ក្បួននិងឬវិធីមួយចំនួន។ ធ្វើដូចនេះ នៅពេលដែលរលកសញ្ញានោះត្រូវផ្សាយចេញ ក្បួន និងឬវិធីទាំងនោះនឹងត្រូវយកទៅប្រើជាស្វ័យប្រវត្តិដែលជាប្រការធ្វើឲ្យបញ្ហាមួយចំនួននឹងត្រូវ យកទៅអនុវត្ត។ ក្បួននិងឬវិធីដែលត្រូវយកទៅតភ្ជាប់ទៅនឹងរលកសញ្ញា ត្រូវហៅថា រន្ធដោត (slot) ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("រលកសញ្ញានិងរន្ធដោត")
```

```

សិស្ស.ផ្លាក = QLabel("តក់ៗពេញបំពង់")
សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម")
សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

```

```

សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.បូកគន្លឹះ)

```

```

សិស្ស.ដង = 0

```

```

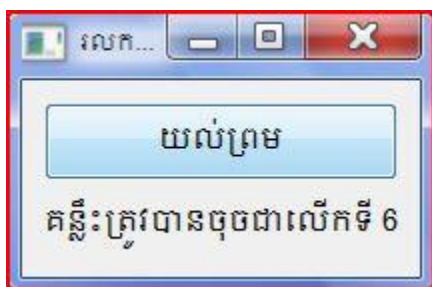
def បូកគន្លឹះ(សិស្ស):
    សិស្ស.ដង += 1
    សិស្ស.ផ្លាក.setText("គន្លឹះត្រូវបានបូកជាលើកទី {0}".format(សិស្ស.ដង))

```

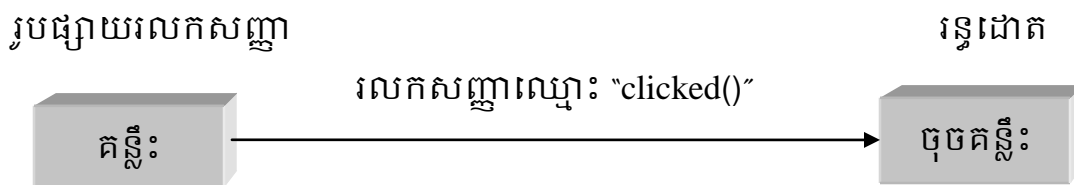
```

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



`សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ)` គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីតភ្ជាប់រលកសញ្ញាឈ្មោះ "clicked()" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ចុចគន្លឹះ ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ នៅពេលណារូបចុងក្រោយនេះត្រូវបានចុច។



ក្រៅពីការតភ្ជាប់រលកសញ្ញាមួយទៅនឹងរន្ធដោតមួយ យើងក៏អាចតភ្ជាប់រលកសញ្ញាជាច្រើន ផ្សាយចេញពីរូបខុសៗគ្នាទៅនឹងរន្ធដោតតែមួយបានដែរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle(" រលកសញ្ញានិងរន្ធដោត")
        សិស្ស.ផ្ទាំង = QLabel(" តក់ៗពេញបំពង់")
        សិស្ស.គន្លឹះ០ = QPushButton("យល់ព្រម")
        សិស្ស.គន្លឹះ១ = QPushButton("បដិសេធ")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ០)
```

សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ១)

សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង)

សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

សិស្ស.connect(សិស្ស.គន្លឹះ០, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ០)

សិស្ស.connect(សិស្ស.គន្លឹះ១, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ១)

សិស្ស.ដង = 0

def ចុចគន្លឹះ១(សិស្ស):

សិស្ស.ដង += 1

សិស្ស.ផ្ទាំង.setText("គន្លឹះត្រូវបានចុចជាលើកទី {0}".format(សិស្ស.ដង))

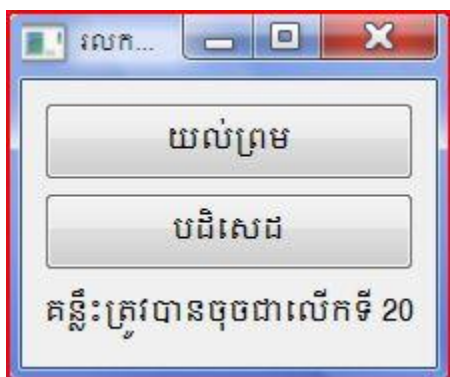
កម្មវិធី = QApplication(sys.argv)

បង្កើត = រូបរមេ()

កម្មវិធី.setFont(QFont('Khmer OS', 11))

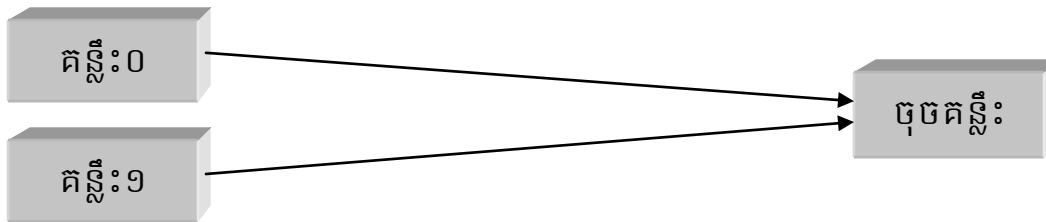
បង្កើត.show()

កម្មវិធី.exec_()



`សិស្ស.connect(សិស្ស.គន្លឹះ០, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ០)` គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីតភ្ជាប់រលកសញ្ញាឈ្មោះ "clicked()" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ចុចគន្លឹះ ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ០ នៅពេលណាដែលប្រើប្រាស់ត្រូវបានចុច។

`សិស្ស.connect(សិស្ស.គន្លឹះ១, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ១)` គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីតភ្ជាប់រលកសញ្ញាឈ្មោះ "clicked()" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ចុចគន្លឹះ ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ១ នៅពេលណាដែលប្រើប្រាស់ត្រូវបានចុច។



ក្រៅពីការតភ្ជាប់រលកសញ្ញាដែលផ្សាយចេញពីរូបចំនួនពីរទៅនឹងរន្ធដោតដែលជាវិធីតែមួយ ដូចគ្នា ផ្ទុយមកវិញ យើងក៏អាចតភ្ជាប់រលកសញ្ញាផ្សាយចេញពីរូបណាមួយទៅនឹងរន្ធដោតពីរ ផ្សេងគ្នាបានដែរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
  
```

```

សិស្ស.setWindowTitle("រលកសញ្ញានិងរន្ធដោត")
សិស្ស.ផ្ទាំង០ = QLabel()
សិស្ស.ផ្ទាំង១ = QLabel()
សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម")
សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង០)
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង១)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.បុប្ផគន្លឹះ០)
សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.បុប្ផគន្លឹះ១)

```

```

សិស្ស.ដង = 0
def បុប្ផគន្លឹះ០(សិស្ស):
    សិស្ស.ដង += 1
    សិស្ស.ផ្ទាំង០.setText("គន្លឹះត្រូវបានបុប្ផជាលើកទី {0}".format(សិស្ស.ដង))

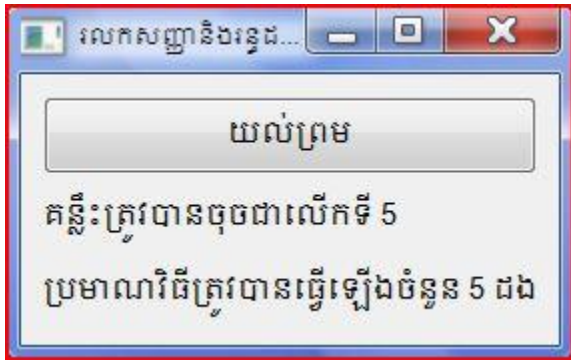
def បុប្ផគន្លឹះ១(សិស្ស):
    សិស្ស.ផ្ទាំង១.setText("ប្រមាណវិធីត្រូវបានធ្វើឡើងចំនួន {0} ដង".format(សិស្ស.ដង))

```

```

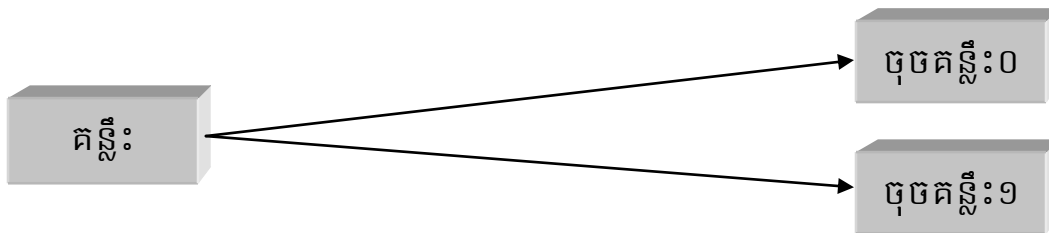
កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



`សិស្ស.connect(សិស្ស.គន្លឹះ០, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ០)` គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីតភ្ជាប់រលកសញ្ញាឈ្មោះ "clicked()" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ចុចគន្លឹះ០ ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ នៅពេលណារូបនេះត្រូវបានចុច។

`សិស្ស.connect(សិស្ស.គន្លឹះ១, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ១)` គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីតភ្ជាប់រលកសញ្ញាឈ្មោះ "clicked()" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ចុចគន្លឹះ១ ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ នៅពេលណារូបនេះត្រូវបានចុច។



ការផ្សាយ រលកសញ្ញាថ្មី

ក្រៅពីលកសណ្ឋានមានស្រាប់ទាំងប៉ុន្មានដែលកើតមានឡើងទាក់ទងទៅនឹងការចុចរូបផ្សេងៗ យើងក៏អាចផ្សាយលកសណ្ឋានថ្មីផ្សេងៗទៀតបានដែរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("លកសណ្ឋាននិងរន្ធដោត")
        សិស្ស.ផ្លាក = QLabel()
        សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

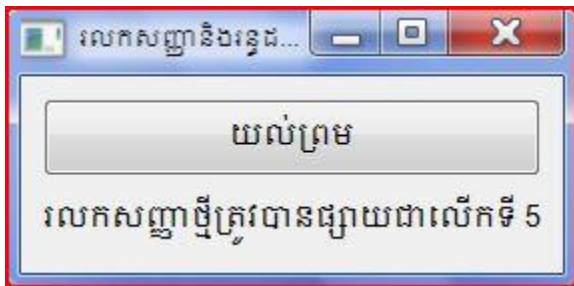
        សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ)
        សិស្ស.connect(សិស្ស, SIGNAL("NewSignal"), សិស្ស.ព័ត៌មាន)

        សិស្ស.ដង = 0
    def ចុចគន្លឹះ(សិស្ស):
        សិស្ស.emit(SIGNAL("NewSignal"))

    def ព័ត៌មាន(សិស្ស):
        សិស្ស.ដង += 1
```

សិស្ស.ផ្ទាំង.setText("រលកសញ្ញាថ្មីត្រូវបានផ្សាយជាលើកទី {0}".format(សិស្ស.ដង))

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបរមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

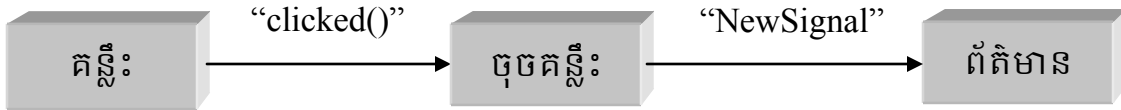


សិស្ស.emit(SIGNAL("NewSignal")) គឺជាការយកវិធីឈ្មោះ emit មកប្រើដើម្បីផ្សាយរលកសញ្ញាថ្មីមានឈ្មោះថា "NewSignal" ។

សិស្ស.connect(សិស្ស, SIGNAL("NewSignal"), សិស្ស.ព័ត៌មាន) គឺជាការយកវិធីឈ្មោះ connect មកប្រើដើម្បីភ្ជាប់រលកសញ្ញាឈ្មោះ "NewSignal" ទៅនឹងរន្ធដោតដែលជាវិធីឈ្មោះ ព័ត៌មាន ។ រលកសញ្ញានេះគឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញពីបង្អួចឈ្មោះ សិស្ស ។

ដូចនេះនៅពេលដែលកម្មវិធីខាងលើកំពុងតែដំណើរការ បើយើងចុចគន្លឹះមានឈ្មោះថា គន្លឹះ រលកសញ្ញាមានស្រាប់មានឈ្មោះថា "clicked()" នឹងត្រូវផ្សាយចេញ។ ប្រការនេះនឹងធ្វើឲ្យវិធីឈ្មោះ ចុចគន្លឹះ នឹងត្រូវយកមកប្រើ ព្រោះវាត្រូវបានភ្ជាប់ទៅនឹងរលកសញ្ញានោះ។ ហើយនៅពេលដែលវិធីឈ្មោះ ចុចគន្លឹះ ត្រូវយកទៅប្រើ រលកសញ្ញាថ្មីមានឈ្មោះថា "NewSignal" ក៏

ត្រូវផ្សាយចេញដែរ។ ហើយនៅពេលដែលរលកសញ្ញាចុងក្រោយនេះត្រូវផ្សាយចេញ វិធី
ឈ្មោះ ព័ត៌មាន នឹងត្រូវយកមកប្រើជាស្វ័យប្រវត្តិ ព្រោះវាត្រូវបានភ្ជាប់ទៅនឹងរលកសញ្ញាថ្មី
នោះ។



រលកសញ្ញាមានដំណឹង

រលកសញ្ញាដែលយើងបានឃើញកន្លងមក គឺជារលកសញ្ញាទទេដែលត្រូវផ្សាយចេញទៅ
ដោយគ្មានព័ត៌មានឬដំណឹងអ្វីនៅជាប់ជាមួយនឹងវាឡើយ។ ក៏ប៉ុន្តែ ចំពោះរលកសញ្ញាមួយ
ចំនួនទៀត នៅពេលដែលវាត្រូវផ្សាយចេញ ព័ត៌មានឬដំណឹងមួយចំនួនក៏ត្រូវផ្សាយចេញ
ទៅជាមួយនឹងវាដែរ។ រលកសញ្ញាទាំងនោះត្រូវចាត់ទុកថាជារលកសញ្ញាមានដំណឹងដែល
យើងអាចយកមកកែច្នៃសម្រាប់យកទៅប្រើការផ្សេងៗ ទៅតាមសេចក្តីត្រូវការនៃការដោះ
ស្រាយបញ្ហារបស់យើង។ ក្រៅពីរលកសញ្ញាមានដំណឹងដែលជារលកសញ្ញាមានស្រាប់ទាំង
នោះ យើងក៏អាចបង្កើតរលកសញ្ញាថ្មីៗដែលមានដំណឹងផ្សេងៗនៅក្នុងនោះបានដែរ
។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
  
```

```

សិស្ស.setWindowTitle(" រលកសញ្ញានិងរន្ធដោត")
សិស្ស.ផ្ទាំង = QLabel()
សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម")
សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("clicked()"), សិស្ស.ចុចគន្លឹះ)
សិស្ស.connect(សិស្ស, SIGNAL("NewSignal"), សិស្ស.ព័ត៌មាន)

```

```

សិស្ស.ដង = 0

```

```

def ចុចគន្លឹះ(សិស្ស):

```

```

    សិស្ស.ដង += 1

```

```

    សិស្ស.emit(SIGNAL("NewSignal"), សិស្ស.ដង)

```

```

def ព័ត៌មាន(សិស្ស, ដង):

```

```

    សិស្ស.ផ្ទាំង.setText(" ដំណឹងដែលត្រូវបានផ្សាយចេញគឺជាលេខ {0}".format(ដង))

```

```

កម្មវិធី = QApplication(sys.argv)

```

```

បង្កើត = រូបរមេ()

```

```

កម្មវិធី.setFont(QFont('Khmer OS', 11))

```

```

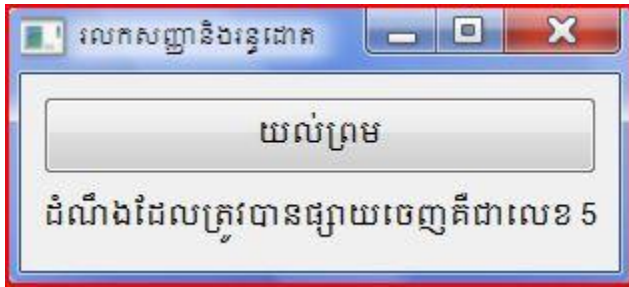
បង្កើត.show()

```

```

កម្មវិធី.exec_()

```



`សិស្ស.emit(SIGNAL("NewSignal"), សិស្ស.ដង)` គឺជាការប្រើវិធីឈ្មោះ emit ដើម្បីផ្សាយរលកសញ្ញាថ្មីមានឈ្មោះថា "NewSignal" និងដំណឹងមួយដែលជាសម្បត្តិសិស្សមានឈ្មោះថា ដង ។

def ព័ត៌មាន(សិស្ស, ដង): គឺជាការបង្កើតវិធីមួយមានឈ្មោះថា ព័ត៌មាន ដែលនឹងត្រូវយកទៅប្រើជាស្វ័យប្រវត្តិនៅពេលដែលរលកសញ្ញាមានឈ្មោះថា "NewSignal" ត្រូវបានផ្សាយចេញ។ ដំណាងដែលជាឈ្មោះ ដង នៅក្នុងវិធីនេះគឺត្រូវប្រើសម្រាប់ទទួលយកដំណឹងដែលជាត្រូវបានផ្សាយចេញមកជាមួយនឹងរលកសញ្ញាឈ្មោះ "NewSignal" នោះ។

ហេតុការណ៍

វិធីឆ្លើយតបហេតុការណ៍

ហេតុការណ៍ក៏មានលក្ខណៈស្រដៀងនឹងរលកសញ្ញាដែរ។ ពោលគឺវាជារឿងហេតុទាំងឡាយណាដែលកើតមានឡើងនៅពេលដែលយើងចុចផ្នែករឹងផ្សេងៗ ដែលទាក់ទងនឹងរូបនៅក្នុងប្រព័ន្ធបង្អួចនៃកម្មវិធី PyQt4 ។ ជាក់ស្តែង ការចុចខ្ទង់ Esc បង្អួចមានហេតុការណ៍មួយប្រភេទកើតមានឡើង។ ហើយដូចគ្នាទៅនឹងរលកសញ្ញាដែរ នៅពេលដែលហេតុការណ៍ណាមួយកើតមានឡើង វិធីណាមួយដែលទាក់ទងទៅនឹងហេតុការណ៍នោះ នឹងត្រូវយកទៅប្រើជា

ស្វ័យប្រវត្តិ។ តែបើគ្មានវិធីណាមួយដែលទាក់ទងទៅនឹងហេតុនោះទេ ហេតុការណ៍នោះនឹងត្រូវបាត់ទៅវិញដោយឯកឯង ដោយគ្មានការផ្លាស់ប្តូរអ្វីឡើយ។ វិធីដែលទាក់ទងទៅនឹងហេតុការណ៍ត្រូវហៅថា *វិធីឆ្លើយតបហេតុការណ៍* (event handler) ដែលភាគច្រើនត្រូវបានបង្កើតឡើងរួចជាស្រេចនៅក្នុងកញ្ចប់ PyQt4 ។ ឧទាហរណ៍ការចុចខ្ទង់ណាមួយនៅលើខ្ទង់កំពូលទំរ នឹងបណ្តាលឲ្យវិធីមានឈ្មោះថា keyPressEvent ត្រូវយកមកប្រើជាស្វ័យប្រវត្តិ។ ដូចនេះ បើយើងចង់ឲ្យវិធីផ្សេងទៀតត្រូវយកទៅប្រើជាស្វ័យប្រវត្តិជំនួសឲ្យវិធីឈ្មោះ keyPressEvent នៅក្នុងកញ្ចប់ PyQt4 យើងត្រូវបង្កើតវិធីមានឈ្មោះដូចគ្នាមួយទៀតបាំងវិធីមានឈ្មោះថា keyPressEvent នេះ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

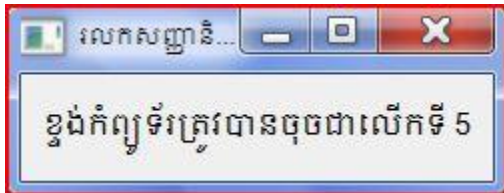
```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("រលកសញ្ញានិងរន្ធដោត")
        សិស្ស.ផ្ទាំង = QLabel()
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)
        សិស្ស.ដង = 0

    def keyPressEvent (សិស្ស, QKeyEvent):
        សិស្ស.ដង += 1
```

សិស្ស.ផ្ទាំង.setText("ខ្ញុំកំពុងត្រូវបានចុចជាលើកទី {0}".format(សិស្ស.ដង))

```
កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបរមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
```



def keyPressEvent (សិស្ស, QKeyEvent): គឺជាការបង្កើតវិធីបំបាត់គេមានឈ្មោះថា keyPressEvent ដែលនឹងត្រូវយកទៅប្រើជាស្វ័យប្រវត្តិដើម្បីឆ្លើយតបទៅនឹងហេតុការណ៍ ដែលកើតមានឡើងបណ្តាលមកពីការចុចខ្នងណាមួយនៅលើខ្នងកំពូលទំរ។

ការស្វែងរកវិធីឆ្លើយតបហេតុការណ៍

រូបទាំងឡាយដែលមាននៅក្នុងប្រព័ន្ធបង្កើត អាចមានទំនាក់ទំនងជាមួយនឹងហេតុការណ៍ផ្សេងៗ ដែលអាចកើតមានឡើងដោយសារការចុចណាខ្នងណាមួយនៅលើខ្នងកំពូលទំរ។ ក្នុងករណីនេះ យើងនិយាយថារូបទាំងនោះស្ថិតក្នុងស្ថានភាព **អាចទទួលបានភ្នែកលើខ្នង** (keyboard focus) ។ បានន័យថា ការចុចខ្នងដែលមានទំនាក់ទំនងជាមួយនឹងរូបទាំងនោះ នឹងបណ្តាលឲ្យការស្វែងរកវិធីឆ្លើយតបហេតុការណ៍ ជាដំបូងត្រូវធ្វើឡើងនៅក្នុងថ្នាក់របស់រូបទាំងនោះជាមុនសិន រួចបានឡើងទៅថ្នាក់របស់រូបផ្សេងៗទៀតដែលមានរូបទាំងនោះរហូតដល់រូបមេធំជាងគេបំផុត។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class គន្លឹះខ្មែរ(QPushButton):
    def __init__(សិស្ស, ផ្លាក):
        QPushButton.__init__(សិស្ស, ផ្លាក)
        សិស្ស.ផ្លាក = QLabel()
        សិស្ស.ដង = 0

    def keyPressEvent (សិស្ស, QKeyEvent):
        សិស្ស.ដង += 1
        សិស្ស.ផ្លាក.setText(" ខ្ញុំ កំពុង ត្រូវបាន ចុច ជា លើក ទី {0} ".format(សិស្ស.ដង))

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle(" លកសញ្ញានិងរន្ធដោត ")
        សិស្ស.គន្លឹះ = គន្លឹះខ្មែរ(" យល់ព្រម ")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))

```

```
បង្ហាញ.show()
កម្មវិធី.exec_()
```

`def keyPressEvent (សិស្ស, QKeyEvent):` គឺជាការបង្កើតវិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា `keyPressEvent` ក្នុងគោលបំណងបំបាត់វិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះដូចគ្នា ដែលត្រូវបានបង្កើតឡើងរួចជាស្រេចនៅក្នុងកញ្ចប់ PyQt4 ។

នៅពេលដែលកម្មវិធីខាងលើនេះកំពុងតែដំណើរការ បើយើងចុចខ្នងណាមួយ ហេតុការណ៍ប្រភេទ `QKeyEvent` នឹងកើតមានឡើង។ ប្រការនេះនឹងធ្វើឲ្យវិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា `keyPressEvent` នឹងត្រូវយកមកប្រើជាស្វ័យប្រវត្តិ។ ហើយការស្វែងរកវិធីនេះ នឹងត្រូវធ្វើឡើងជាដំបូងនៅក្នុងថ្នាក់របស់គន្លឹះដែលមានឈ្មោះថា គន្លឹះ ព្រោះគន្លឹះនោះស្ថិតនៅក្នុងស្ថានភាពអាចទទួលបានទិពលខ្លាំង។ បើសិនជានៅក្នុងថ្នាក់របស់គន្លឹះនោះគ្មានវិធីណាមួយមានឈ្មោះថា `keyPressEvent` ទេ ការស្វែងរកវិធីមានឈ្មោះដូចនេះនឹងត្រូវធ្វើជាបន្តទៅទៀតនៅក្នុងថ្នាក់នៃរូបមេដែលមានគន្លឹះនោះ ដែលជាថ្នាក់មានឈ្មោះថា រូបមេ ។ វិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា `keyPressEvent` នៅក្នុងថ្នាក់ឈ្មោះ គន្លឹះខ្មែរ នឹងត្រូវយកមកប្រើព្រោះវានឹងត្រូវរកឃើញនៅទីនោះមុនគេ។

វិធីឆ្លើយតបហេតុការណ៍ឈ្មោះ event

យើងបានដឹងរួចមកហើយថា នៅពេលណាដែលខ្លាំងណាមួយត្រូវបានចុច វិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា `keyPressEvent` នឹងត្រូវយកមកប្រើជាស្វ័យប្រវត្តិ។ ក៏ប៉ុន្តែជាទូទៅ នៅពេលដែលហេតុការណ៍ណាមួយកើតមានឡើង វិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា `event` នឹងត្រូវយកមកប្រើមុនគេបង្អស់ មុនវិធីឆ្លើយតបហេតុការណ៍ផ្សេងៗទៀត។ ដូចនេះ បើសិនណាជាយើងបង្កើតវិធីមួយមានឈ្មោះថា `event` នៅក្នុងកម្មវិធីរបស់យើង វិធីនេះនឹងត្រូវយក

ទៅប្រើជាស្វ័យប្រវត្តិ នៅរាល់ពេលដែលហេតុការណ៍ទាំងឡាយកើតមានឡើង។ ពិនិត្យ
កម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class គន្លឹះខ្មែរ(QPushButton):
    def __init__(សិស្ស, ផ្លាក):
        QPushButton.__init__(សិស្ស, ផ្លាក)
        សិស្ស.ផ្លាក = QLabel()
        សិស្ស.ដង = 0

    def event (សិស្ស, ហេតុការណ៍):
        សិស្ស.ដង += 1
        សិស្ស.ផ្លាក.setText(" ខ្ញុំកំពុងត្រូវបានចុចជាលើកទី {0} ".format(សិស្ស.ដង))

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle(" រលកសញ្ញានិងរន្ធដោត")
        សិស្ស.គន្លឹះ = គន្លឹះខ្មែរ("យល់ព្រម")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)
```

```
កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
```

def event (សិស្ស, ហេតុការណ៍): គឺជាការបង្កើតវិធីឆ្លើយតបហេតុការណ៍មានឈ្មោះថា event ដែលនឹងត្រូវយកទៅប្រើជាស្វ័យប្រវត្តិ មុនវិធីឆ្លើយតបហេតុការណ៍ផ្សេងៗទៀត នៅរាល់ពេលដែលហេតុការណ៍ណាមួយកើតមានឡើង។

រូបនៅក្នុងប្រព័ន្ធបង្កាប QT4

នៅក្នុងប្រព័ន្ធបង្កាប QT4 មានរូបជាច្រើនប្រភេទដែលអាចត្រូវបង្កើតឡើងដើម្បីឆ្លើយតបទៅនឹងសេចក្តីត្រូវការផ្សេងៗនៅក្នុងការសរសេរកម្មវិធីដែលមានលក្ខណៈជាប្រព័ន្ធបង្កាប (GUI) ។ រូបទាំងនោះមានដូចតទៅនេះ៖

គន្លឹះ

ការបង្កើតគន្លឹះ

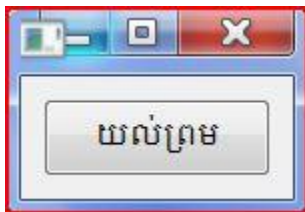
យើងបានឃើញរួចមកហើយថា **គន្លឹះ** (button) គឺជារូបម្យ៉ាងនៅក្នុងប្រព័ន្ធបង្កាប QT4 ដែលមានរាងជាចតុកោណកែងតូចមួយប្រើសម្រាប់ចុចបង្កើតរលកសញ្ញាមានឈ្មោះថា “clicked()” ។ ហើយដើម្បីបង្កើតរូបដែលជាគន្លឹះនោះ យើងត្រូវយកថ្នាក់ឈ្មោះ QPushButton មកប្រើដោយធ្វើដូចខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះ")
        សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
```

```
សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)
```

```
កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបរមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
```



សិស្ស.គន្លឹះ = QPushButton("យល់ព្រម") គឺជាការយកថ្នាក់ឈ្មោះ QPushButton មកប្រើ ដើម្បីបង្កើតគន្លឹះមួយមានឈ្មោះថា គន្លឹះ ដែលសម្បត្តិសិស្ស។

ស្ថាបនិកនៃថ្នាក់ QPushButton

ស្ថាបនិកនៃថ្នាក់ QPushButton មានទម្រង់ដូចខាងក្រោមនេះ៖

- __init__(self, QWidget parent = None)
- __init__(self, QString text, QWidget parent = None)
- __init__(self, QIcon icon, QString text, QWidget parent = None)
 - parent : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជារូបមានប្រភេទជា QWidget ។ ក៏ប៉ុន្តែជាទូទៅយើងមិនចាំបាច់ផ្តល់ដំណឹងអ្វីសម្រាប់ដំណាង parent នេះទេ ព្រោះដំណឹងសម្រាប់ដំណាងនេះនឹងត្រូវផ្តល់ឲ្យជាស្វ័យប្រវត្តិនៅពេលណា ដែលគន្លឹះត្រូវបង្កើត ត្រូវយកទៅបិទនៅក្នុងរូបមេណាមួយ តាមរយៈក្រឡាទំហំ

ណាមួយ។ ហើយដំណឹងសម្រាប់ដំណាង parent នេះ នឹងជារូបមេដែល

គន្លឹះត្រូវបង្កើត ត្រូវយកទៅបិទនៅក្នុងនោះ។

- **text** : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាកម្រងអក្សរត្រូវយកទៅសរសេរនៅលើគន្លឹះត្រូវបង្កើត។
- **icon** : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជា *រូបដំណាង* (icon) ណាមួយដែលត្រូវយកទៅបិទនៅលើគន្លឹះត្រូវបង្កើត។

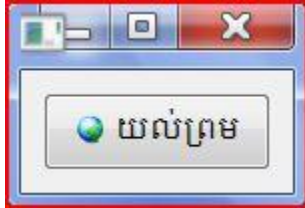
ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះ")
        សិស្ស.គន្លឹះ = QPushButton(QIcon("icons\Globe"), "យល់ព្រម")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
```

កម្មវិធី.exec_()



សិស្ស.គន្លឹះ = QPushButton(QIcon("icons\Globe"), "យល់ព្រម") គឺជាការយកថ្នាក់ឈ្មោះ QPushButton មកប្រើដើម្បីបង្កើតគន្លឹះមួយមានឈ្មោះថា គន្លឹះ ដែលជាសម្បត្តិសិស្ស។ ដំណឹងដែលជា QIcon("icons\Globe") គឺជារូបដំណាងដែលត្រូវយកទៅបិទនៅលើគន្លឹះមានឈ្មោះថា គន្លឹះ នោះ។ រូបដំណាងនោះគឺជារូបដំណាងមានឈ្មោះថា Globe ស្ថិតនៅក្នុងថតឈ្មោះ icons ។ ចំណែកដំណឹងដែលជាកម្រងអក្សរ "យល់ព្រម" វិញ គឺសម្រាប់យកទៅសរសេរនៅលើគន្លឹះមានឈ្មោះថា គន្លឹះ នោះ។

វិធីសំខាន់ៗនៃថ្នាក់ QPushButton

ដោយថ្នាក់ឈ្មោះ QPushButton គឺជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ QAbstractButton ដូចនេះ តាមរយៈសិស្សនៃថ្នាក់ឈ្មោះ QPushButton យើងអាចយកវិធីទាំងអស់នៅក្នុងឈ្មោះ QAbstractButton មកប្រើបានតាមសេចក្តីត្រូវការ។ វិធីសំខាន់ៗនៅក្នុងថ្នាក់ឈ្មោះ QAbstractButton មានដូចតទៅនេះ៖

- **setText (self, QString text)** : គឺជាវិធីប្រើសម្រាប់កំណត់យកកម្រងអក្សរណាមួយជាកម្រងអក្សរដែលត្រូវសរសេរនៅលើគន្លឹះណាមួយ។
- **setIcon (self, QIcon icon)** : គឺជាវិធីប្រើសម្រាប់កំណត់យករូបដំណាងណាមួយជារូបដំណាងដែលត្រូវយកទៅបិទនៅលើគន្លឹះណាមួយ។

- `setCheckable (self, bool)` : គឺជាវិធីប្រើសម្រាប់ធ្វើឲ្យគន្លឹះណាមួយអាចស្ថិតនៅក្នុងស្ថានពីរគឺ ស្ថានភាពត្រូវបិទ និងស្ថានភាពមិនត្រូវបិទ (toggle button) ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះ")
        សិស្ស.គន្លឹះ = QPushButton(QIcon("icons\Globe"), "យល់ព្រម")
        សិស្ស.គន្លឹះ.setCheckable (True)
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
```



សិស្ស.គន្លឹះ::setChecked (True) គឺជាការយកវិធីឈ្មោះ setCheckable មកប្រើដើម្បីធ្វើឲ្យគន្លឹះ មានឈ្មោះថា គន្លឹះ អាចត្រូវស្ថិតនៅក្នុងស្ថានភាពត្រូវចុចនិងមិនត្រូវចុច។

- bool isChecked (self) : គឺជាវិធីប្រើសម្រាប់ពិនិត្យមើលស្ថានភាពរបស់គន្លឹះណាមួយថា តើវាស្ថិតនៅក្នុងស្ថានភាពត្រូវចុចមែនដែរឬទេ។ វិធីនេះត្រូវយកទៅប្រើជាមួយតែគន្លឹះ ណាដែលអាចត្រូវស្ថិតក្នុងស្ថានភាពត្រូវចុចនិងមិនត្រូវចុចតែប៉ុណ្ណោះ។

រលកសញ្ញា

រលកសញ្ញាសំខាន់ៗដែលអាចត្រូវផ្សាយចេញទាក់ទងទៅនឹងរូបដែលជាគន្លឹះ គឺជារលកសញ្ញាមានដូចខាងក្រោមនេះ៖

- “clicked ()” : គឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញនៅពេលដែលគន្លឹះត្រូវបានចុច។
- “toggled (bool)” : គឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញនៅពេលគន្លឹះដែលអាចត្រូវស្ថិត នៅក្នុងស្ថានភាពត្រូវចុចនិងមិនត្រូវចុច ត្រូវផ្លាស់ប្តូរស្ថានភាព។ ម៉្យាងទៀត នៅពេល ដែលរលកសញ្ញានេះត្រូវផ្សាយចេញពីគន្លឹះណាមួយ ដំណឹងដែលជាតក្កវត្ថុដំណាងឲ្យ ស្ថានភាពរបស់គន្លឹះនោះ ក៏ត្រូវផ្សាយចេញទៅជាមួយនឹងរលកសញ្ញានោះដែរ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
```

```

from PyQt4.QtGui import*

class រូបរាង(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះ")
        សិស្ស.គន្លឹះ = QPushButton(QIcon("icons\Globe"), "យល់ព្រម")
        សិស្ស.ផ្លាក = QLabel()
        សិស្ស.គន្លឹះ.setCheckable (True)
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.គន្លឹះ)
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

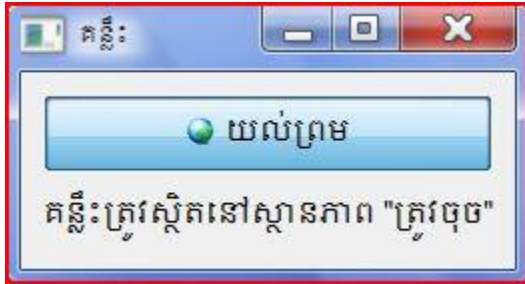
        សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("toggled(bool)"), សិស្ស.បូកគន្លឹះ)

    def បូកគន្លឹះ(សិស្ស, ស្ថានភាព):
        if ស្ថានភាព :
            ស្ថានភាព = "ត្រូវបូក"
        else:
            ស្ថានភាព = "មិនត្រូវបូក"
        សិស្ស.ផ្លាក.setText(' គន្លឹះត្រូវស្ថិតនៅស្ថានភាព "{0}"'.format(ស្ថានភាព))

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបរាង()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()

```

កម្មវិធី.exec_()



សិស្ស.connect(សិស្ស.គន្លឹះ, SIGNAL("toggled(bool)"), សិស្ស.ចុចគន្លឹះ) គឺជាការតភ្ជាប់រលកសញ្ញាឈ្មោះ "toggled(bool)" ដែលត្រូវផ្សាយចេញពីគន្លឹះឈ្មោះ គន្លឹះ ទៅនឹងរន្ធដោតឈ្មោះ ចុចគន្លឹះ ។ នៅក្នុងរលកសញ្ញានេះមានដំណឹងមានប្រភេទជាតក្កវត្ថុមួយ ដែលដំណាងឲ្យស្ថានភាពរបស់គន្លឹះនោះថាវាត្រូវស្ថិតក្នុងស្ថានភាពត្រូវចុចឬមិនត្រូវចុច។

def ចុចគន្លឹះ(សិស្ស, ស្ថានភាព): គឺជាការបង្កើតវិធីមួយដែលជារន្ធដោតឈ្មោះ ចុចគន្លឹះ ។ នៅក្នុងវិធីនេះមានដំណាងដែលជាឈ្មោះ ស្ថានភាព សម្រាប់ទទួលយកដំណឹងដែលនឹងត្រូវផ្សាយចេញមកជាមួយនឹងរលកសញ្ញាផ្សេងៗ។

គន្លឹះជម្រើស

ការបង្កើតគន្លឹះជម្រើស

គន្លឹះជម្រើស (radio button) គឺជារូបម៉្យាងប្រើសម្រាប់ធ្វើការជ្រើសរើសយកជម្រើសមួយនៅក្នុងចំណោមជម្រើសជាច្រើន។ ដើម្បីបង្កើតគន្លឹះជម្រើស យើងត្រូវយកថ្នាក់ឈ្មោះ QRadioButton មកប្រើ ដោយសរសេរកម្មវិធីដូចខាងក្រោមនេះ៖

```
import sys
```

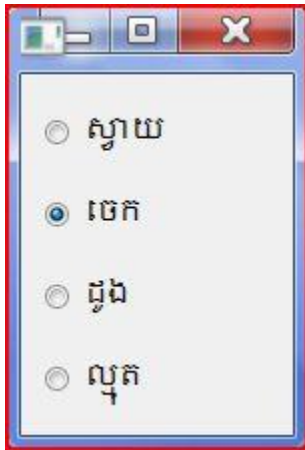
```

from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះជម្រើស")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្ទៃឈើ = []
        for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
            សិស្ស.ផ្ទៃឈើ.append(QRadioButton(ផ្ទៃ))
            សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1])
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



សិស្ស.ផ្ទៃឈើ.append(QRadioButton(ផ្ទៃ)) គឺជាបញ្ហាដែលនៅក្នុងនោះមានការតម្រូវឲ្យយក ថ្នាក់ឈ្មោះ: QRadioButton មកប្រើដើម្បីបង្កើតគន្លឹះជម្រើសដែលជាធាតុនៅក្នុងកម្រងអថេ ឈ្មោះ: ផ្ទៃឈើ ។

ស្ថាបនិកនៃថ្នាក់ QRadioButton

ស្ថាបនិកនៃថ្នាក់ឈ្មោះ: QRadioButton មានទម្រង់ដូចខាងក្រោមនេះ៖

- __init__(self, QWidget parent = None)
- __init__(self, QString text, QWidget parent = None)
 - text : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាកម្រងអក្សរត្រូវយកទៅសរសេរ នៅក្បែរគន្លឹះជម្រើស។

វិធីសំខាន់ៗនៃថ្នាក់ QRadioButton

យើងសង្កេតឃើញថា គន្លឹះជម្រើសដែលស្ថិតនៅក្នុងក្រឡាទំហំជាមួយគ្នា មានលក្ខណៈ

មានមួយអត់មួយ (mutually exclusive) ។ បានន័យថា នៅពេលណាដែលគន្លឹះជម្រើសណា

មួយត្រូវបានជ្រើសរើសយក គន្លឹះជម្រើសផ្សេងៗទៀតនឹងលែងត្រូវបានជ្រើសរើសយក។
ប្រការនេះធ្វើឲ្យយើងអាចជ្រើសរើសយកគន្លឹះជម្រើសម្តងបានតែមួយតែប៉ុណ្ណោះ។

ម៉្យាងទៀត ដោយថ្នាក់ឈ្មោះ QRadioButton ជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ QAbstractButton
ដូចនេះ យើងអាចយកវិធីទាំងអស់ដែលមាននៅក្នុងថ្នាក់ឈ្មោះ QAbstractButton មកប្រើ
ជាមួយនឹងគន្លឹះជម្រើសទាំងឡាយណាដែលជាសិស្សនៃថ្នាក់ឈ្មោះ QRadioButton ។ ពិនិត្យ
កម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះជម្រើស")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្ទៃឈើ = []
        for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
            សិស្ស.ផ្ទៃឈើ.append(QRadioButton(ផ្ទៃ))
            សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1].setIcon(QIcon("icons\Globe"))
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1])
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
```

```

កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្ហាញ.show()
កម្មវិធី.exec_()

```



សិស្ស.ផ្ទៃលើ[len(សិស្ស.ផ្ទៃលើ) - 1].setIcon(QIcon("icons\Globe")) គឺជាបញ្ហាដែលនៅក្នុងនោះមានការតម្រូវឲ្យយកវិធីឈ្មោះ setIcon នៅក្នុងថ្នាក់ឈ្មោះ QAbstractButton មកប្រើដើម្បីយករូបដំណាងឈ្មោះ Globe ទៅបិទនៅក្បែរគន្លឹះជម្រើសដែលជាធាតុនៅក្នុងកម្រងអថេរឈ្មោះ ផ្ទៃលើ ។

រលកសញ្ញា

រលកសញ្ញាសំខាន់ៗដែលអាចត្រូវផ្សាយចេញពីគន្លឹះជម្រើស គឺជារលកសញ្ញាមានដូចខាងក្រោមនេះ៖

- “toggled (bool)” : គឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញនៅពេលដែលគន្លឹះជម្រើសផ្លាស់ប្តូរស្ថានភាព។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
```

```

from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("គន្លឹះជម្រើស")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្លាក = QLabel()
        សិស្ស.ផ្ទៃឈើ = []
        for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
            សិស្ស.ផ្ទៃឈើ.append(QRadioButton(ផ្ទៃ))
            សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1].setIcon(QIcon("icons\Globe"))
            សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1])
            សិស្ស.connect(សិស្ស.ផ្ទៃឈើ[len(សិស្ស.ផ្ទៃឈើ) - 1], SIGNAL("toggled(bool)",
                សិស្ស.ចុចគន្លឹះ))

        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

    def ចុចគន្លឹះ(សិស្ស, ស្ថានភាព):
        for ផ្ទៃ in សិស្ស.ផ្ទៃឈើ :
            if ផ្ទៃ.isChecked():
                សិស្ស.ផ្លាក.setText('ផ្ទៃ{0}ត្រូវបានជ្រើសរើសយក។'.format(ផ្ទៃ.text()))

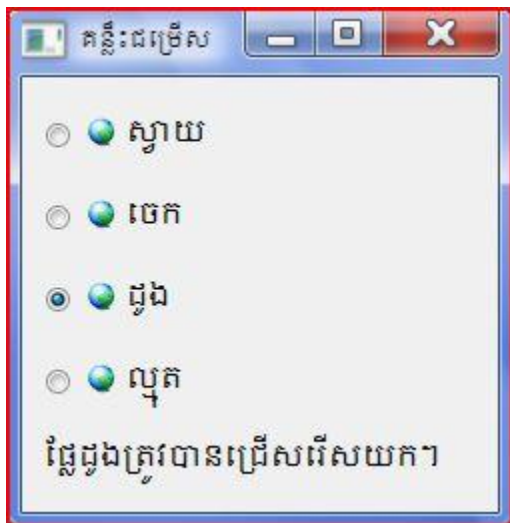
កម្មវិធី = QApplication(sys.argv)

```

```

បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



សិស្ស.connect(សិស្ស.ផ្លែឈើ[len(សិស្ស.ផ្លែឈើ) - 1], SIGNAL("toggled(bool)"),

សិស្ស.ចុចគន្លឹះ) គឺជាការភ្ជាប់រលកសញ្ញាឈ្មោះ "toggled(bool)" ទៅនឹងរន្ធដោយឈ្មោះ ចុចគន្លឹះ នៅពេលដែលគន្លឹះជម្រើសដែលជាធាតុនៅក្នុងកម្រងអថេរឈ្មោះ ផ្លែឈើ ត្រូវបាន ចុច។

ក្រឡាគូស

ការបង្កើតក្រឡាគូស

ក្រឡាគូស (check box) គឺជារូបម្យ៉ាងប្រើសម្រាប់គូសដើម្បីធ្វើការជ្រើសរើសយកអ្វីមួយ។

ដើម្បីបង្កើតក្រឡាគូស យើងត្រូវយកថ្នាក់ឈ្មោះ QCheckBox មកប្រើ ដោយសរសេរកម្មវិធី ដូចខាងក្រោមនេះ៖

```

import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

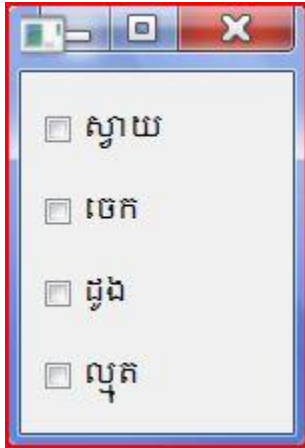
class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ក្រឡាគូស")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្ទៃឈើ = []
        លេខរៀង = 0
        for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
            សិស្ស.ផ្ទៃឈើ.append(QCheckBox(ផ្ទៃ))
            សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[លេខរៀង])
            លេខរៀង += 1

        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្អួច = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្អួច.show()

```

កម្មវិធី.exec_()



សិស្ស.ផ្ទៃឈើ.append(QCheckBox(ផ្ទៃ)) គឺជាបញ្ហាដែលនៅក្នុងនោះមានការតម្រូវឲ្យយកថ្នាក់ឈ្មោះ QCheckBox មកប្រើដើម្បីបង្កើតក្រឡាគូសដែលជាធាតុនៅក្នុងកម្រងអថេរឈ្មោះផ្ទៃឈើ ។

ស្ថាបនិកនៃថ្នាក់ QCheckBox

ស្ថាបនិកនៃថ្នាក់ឈ្មោះ QCheckBox មានទម្រង់ដូចខាងក្រោមនេះ៖

- __init__ (self, QWidget parent = None)
- __init__ (self, QString text, QWidget parent = None)

វិធីសំខាន់ៗនៃថ្នាក់ QCheckBox

ថ្នាក់ឈ្មោះ QCheckBox គឺជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ QAbstractButton ។ ដូចនេះ តាមរយៈក្រឡាគូសដែលជាសិស្សនៃថ្នាក់ឈ្មោះ QCheckBox យើងអាចវិធីទាំងអស់ដែលមាននៅក្នុងថ្នាក់ QAbstractButton មកប្រើបានតាមសេចក្តីត្រូវការ។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
```

```

from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ក្រុងភ្នំគូស")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្ទៃឈើ = []
        លេខរៀង = 0
        for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
            សិស្ស.ផ្ទៃឈើ.append(QCheckBox(ផ្ទៃ))
            សិស្ស.ផ្ទៃឈើ[លេខរៀង].setIcon(QIcon("icons\Globe"))
            សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[លេខរៀង])
            លេខរៀង += 1

        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



សិស្ស.ផ្ទៃឈើលេខរៀង].setIcon(QIcon("icons\Globe")) គឺជាបញ្ហាដែលនៅក្នុងនោះមានការតម្រូវឲ្យយកវិធីឈ្មោះ: setIcon នៅក្នុងថ្នាក់មេឈ្មោះ: QAbstractButton មកប្រើដើម្បីចម្លងយករូបដំណាងឈ្មោះ: Globe ទៅបិទនៅក្បែកក្រឡាគូសដែលជាធាតុនៅក្នុងកម្រងអថេរឈ្មោះ: ផ្ទៃឈើ ។

រលកសញ្ញា

រលកសញ្ញាសំខាន់ៗដែលអាចត្រូវផ្សាយចេញពីក្រឡាគូស គឺជារលកសញ្ញាមានដូចខាងក្រោមនេះ៖

- “stateChanged (int)” : គឺជារលកសញ្ញាដែលត្រូវផ្សាយចេញនៅពេលណាដែលក្រឡាគូសផ្លាស់ប្តូរសភាពរបស់វា។ ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
```

```

QWidget.__init__(សិស្ស)
សិស្ស.setWindowTitle("ក្រុងឡាតូស")
សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
សិស្ស.ផ្ទាំង = QLabel()
សិស្ស.ផ្ទៃឈើ = []
លេខរៀង = 0
for ផ្ទៃ in ["ស្វាយ", "ចេក", "ដូង", "ល្អិត"]:
    សិស្ស.ផ្ទៃឈើ.append(QCheckBox(ផ្ទៃ))
    សិស្ស.ផ្ទៃឈើ[លេខរៀង].setIcon(QIcon("icons\Globe"))
    សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទៃឈើ[លេខរៀង])
    សិស្ស.connect(សិស្ស.ផ្ទៃឈើ[លេខរៀង], SIGNAL("stateChanged(int)"),
                 សិស្ស.ចុចគន្លឹះ)
    លេខរៀង += 1

សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្ទាំង)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

def ចុចគន្លឹះ(សិស្ស, សភាព):
    ផ្ទៃឈើ = " "
    for ផ្ទៃ in សិស្ស.ផ្ទៃឈើ :
        if ផ្ទៃ.isChecked():
            ផ្ទៃឈើ += ផ្ទៃ.text() + ' '
    សិស្ស.ផ្ទាំង.setText('ផ្ទៃ{0}ត្រូវបានជ្រើសរើសយក។.format(ផ្ទៃឈើ))

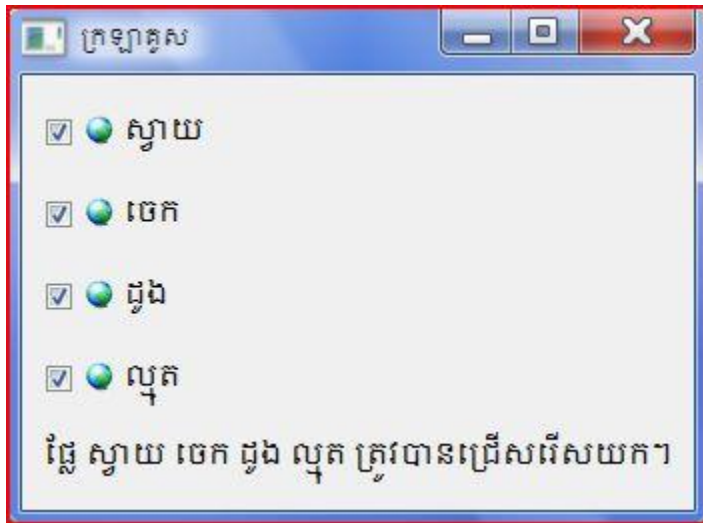
កម្មវិធី = QApplication(sys.argv)

```

```

បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()

```



សិស្ស.connect(សិស្ស.ផ្លែលើលេខរៀង, SIGNAL("stateChanged(int)"), សិស្ស.ប៉ុចគន្លឹះ) គឺ ជាការតភ្ជាប់រលកសញ្ញាឈ្មោះ "stateChanged(int)" ទៅនឹងរន្ធដោតឈ្មោះ ប៉ុចគន្លឹះ នៅពេលដែលក្រឡាគូសដែលជាធាតុនៅក្នុងកម្រងអថេរឈ្មោះ ផ្លែលើ ត្រូវបានប៉ុច។

ផ្លាស់

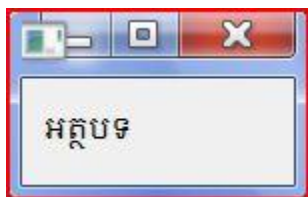
ការបង្កើតផ្លាក

ផ្លាក (label) គឺជារូបមានរាងជាចតុកោណកែង ប្រើសម្រាប់បង្ហាញអត្ថបទប្រូបគំនូរផ្សេងៗ។ ដើម្បីបង្កើតផ្លាក យើងត្រូវយកថ្នាក់ឈ្មោះ QLabel មកប្រើដោយសរសេរកម្មវិធីដូចខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ផ្លាក")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្លាក = QLabel("អត្ថបទ")
        សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
        សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()
```



សិស្ស. ផ្លាក = QLabel("អត្ថបទ") គឺជាការយកផ្លាកឈ្មោះ: QLabel មកប្រើដើម្បីបង្កើតផ្លាក មានឈ្មោះថា ផ្លាក មួយដែលជាសម្បត្តិ។

ស្ថាបនិកនៃផ្លាក QLabel

ស្ថាបនិកនៃផ្លាក QLabel មានទម្រង់ដូចខាងក្រោមនេះ៖

- `__init__(self, QWidget parent = None, Qt.WindowFlags f = 0)`
- `__init__(self, QString text, QWidget parent = None, Qt.WindowFlags f = 0)`
 - text : គឺជាដំណាងចាំទទួលយកដំណឹងដែលជាអត្ថបទត្រូវយកទៅសរសេរនៅលើផ្លាកត្រូវបង្កើត។
 - f : គឺជាដំណាងចាំទទួលយកដំណឹងប្រភេទជា Qt.WindowFlags ។

វិធីសំខាន់ៗនៃផ្លាក QLabel

វិធីសំខាន់ៗនៃផ្លាកឈ្មោះ: QLabel មានដូចខាងក្រោមនេះ៖

- `setText (self, QString)` : គឺជាវិធីប្រើសម្រាប់កំណត់យកអត្ថបទជាអត្ថបទដែលត្រូវសរសេរនៅលើផ្លាកណាមួយ។
- `QString text (self)` : គឺជាវិធីប្រើសម្រាប់ចម្លងយកអត្ថបទដែលមាននៅក្នុងផ្លាកណាមួយ។

ដោយហេតុថាថ្នាក់ឈ្មោះ QLabel ជាថ្នាក់រងនៃថ្នាក់ឈ្មោះ QFrame ដូចនេះតាមរយៈផ្លាកដែលជាសិស្សនៃថ្នាក់ឈ្មោះ QLabel នេះ យើងអាចវិធីទាំងអស់ដែលមាននៅក្នុងថ្នាក់ឈ្មោះ QFrame នោះមកប្រើបានតាមសេចក្តីត្រូវការ។ ហើយវិធីសំខាន់នៅក្នុងថ្នាក់ឈ្មោះ QFrame មានដូចខាងក្រោមនេះ៖

- `setFrameStyle (self, int)` : គឺជាវិធីប្រើសម្រាប់កំណត់យក *រចនាបទ* (style) ណាមួយជារចនាបទសម្រាប់រូបណាមួយ។ ដំណាង int នៅក្នុងវិធីនេះ គឺជាដំណាងចាំទទួលយករចនាបទទាំងឡាយមានដូចខាងក្រោមនេះ៖
 - `QFrame.Plain` : គឺជារចនាបទដែលធ្វើឲ្យរូបមានផ្ទៃរាបស្មើ
 - `QFrame.Raised` : គឺជារចនាបទដែលធ្វើឲ្យរូបមានផ្ទៃប៉ោងឡើង
 - `QFrame.Sunken` : គឺជារចនាបទដែលធ្វើឲ្យរូបមានផ្ទៃជ្រកចុះ
 - `QFrame.Box` : គឺជារចនាបទដែលធ្វើឲ្យរូបមានវណ្ណរាងចតុកោណកែងព័ទ្ធជុំវិញ

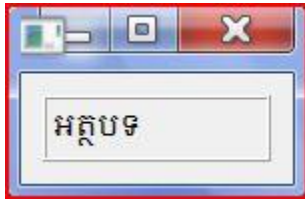
ពិនិត្យកម្មវិធីខាងក្រោមនេះ៖

```
import sys
from PyQt4.QtCore import*
from PyQt4.QtGui import*

class រូបមេ(QWidget):
    def __init__(សិស្ស):
        QWidget.__init__(សិស្ស)
        សិស្ស.setWindowTitle("ផ្លាក")
        សិស្ស.ប្រអប់ទំហំឈរ = QVBoxLayout()
        សិស្ស.ផ្លាក = QLabel("អត្ថបទ")
        សិស្ស.ផ្លាក.setFrameStyle (QFrame.Raised | QFrame.Box)
```

សិស្ស.ប្រអប់ទំហំឈរ.addWidget(សិស្ស.ផ្លាក)
សិស្ស.setLayout(សិស្ស.ប្រអប់ទំហំឈរ)

កម្មវិធី = QApplication(sys.argv)
បង្កើត = រូបរមេ()
កម្មវិធី.setFont(QFont('Khmer OS', 11))
បង្កើត.show()
កម្មវិធី.exec_()



សិស្ស.ផ្លាក.setStyle(QFrame.Raised | QFrame.Box) គឺជាការយកវិធីឈ្មោះ

setFrameStyle នៅក្នុងថ្នាក់ឈ្មោះ QFrame មកប្រើដើម្បីកំណត់យករចនាបទ

QFrame.Raised ផ្តល់ជាមួយនឹងរចនាបទ QFrame.Box ជារចនាបទរបស់ផ្លាកមានឈ្មោះថា
ផ្លាក ។

ដូចនេះ យើងសង្កេតឃើញថា យើងអាចយករចនាបទមួយចំនួនមកផ្តួចផ្តង់ផ្តល់ជាដំណឹងឲ្យ
ទៅដំណាងផ្សេងៗនៅក្នុងវិធីណាមួយបាន ហើយការផ្តួចផ្តង់នេះ គឺត្រូវធ្វើឡើងដោយប្រើ
ប្រមាណសញ្ញា | ។

របារនៃការវិវត្ត

ការបង្កើតរបៀបនៃការវិវត្តន៍