No changes!

#!/usr/bin/env python3

"""PyCalc is a simple calculator built using Python and PyQt5."""

import sys

# Import QApplication and the required widgets from PyQt5.QtWidgets

from PyQt5.QtWidgets import QApplication

from PyQt5.QtWidgets import QMainWindow

from PyQt5.QtWidgets import QWidget

\_\_version\_\_ = **'0.1'**

\_\_author\_\_ = **'Malith Thiwanka'**

# Create a subclass of QMainWindow to setup the calculator's GUI

class PyCalcUi(QMainWindow):

"""PyCalc's View (GUI)."""

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

"""View initializer."""

super().\_\_init\_\_()

# Set some main window's properties

self.setWindowTitle(**'PyCalc'**)

self.setFixedSize(235, 235)

# Set the central widget

self.\_centralWidget = QWidget(self)

self.setCentralWidget(self.\_centralWidget)

# Client code

def main():

"""Main function."""

# Create an instance of QApplication

pycalc = QApplication(sys.argv)

# Show the calculator's GUI

view = PyCalcUi()

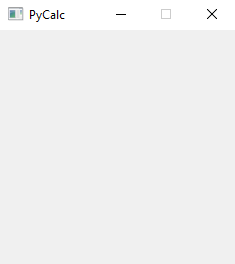
view.show()

# Execute the calculator's main loop

sys.exit(pycalc.exec\_())

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

main()



==========================================================

Changes -> 1 to 5

#!/usr/bin/env python3

"""PyCalc is a simple calculator built using Python and PyQt5."""

import sys

# Import QApplication and the required widgets from PyQt5.QtWidgets

from PyQt5.QtWidgets import QApplication

from PyQt5.QtWidgets import QMainWindow

from PyQt5.QtWidgets import QWidget

# change 01 - start

from PyQt5.QtCore import Qt

from PyQt5.QtWidgets import QGridLayout

from PyQt5.QtWidgets import QLineEdit

from PyQt5.QtWidgets import QPushButton

from PyQt5.QtWidgets import QVBoxLayout

# change 01 -end

\_\_version\_\_ = **'0.1'**

\_\_author\_\_ = **'Malith Thiwanka'**

# Create a subclass of QMainWindow to setup the calculator's GUI

class PyCalcUi(QMainWindow):

"""PyCalc's View (GUI)."""

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

"""View initializer."""

super().\_\_init\_\_()

# Set some main window's properties

self.setWindowTitle(**'PyCalc'**)

self.setFixedSize(235, 235)

# Set the central widget and the general layout

# change 02

self.generalLayout = QVBoxLayout()

self.\_centralWidget = QWidget(self)

self.setCentralWidget(self.\_centralWidget)

# change 03 stat

self.\_centralWidget.setLayout(self.generalLayout)

# Create the display and the buttons

self.\_createDisplay()

self.\_createButtons()

# change 03 - end

# change 04 - start

def \_createDisplay(self):

"""Create the display."""

# Create the display widget

self.display = QLineEdit()

# Set some display's properties

self.display.setFixedHeight(35)

self.display.setAlignment(Qt.AlignRight)

self.display.setReadOnly(True)

# Add the display to the general layout

self.generalLayout.addWidget(self.display)

# change 04 - end

# change 05 - start

def \_createButtons(self):

"""Create the buttons."""

self.buttons = {}

buttonsLayout = QGridLayout()

# Button text | position on the QGridLayout

buttons = {**'7'**: (0, 0),

**'8'**: (0, 1),

**'9'**: (0, 2),

**'/'**: (0, 3),

**'C'**: (0, 4),

**'4'**: (1, 0),

**'5'**: (1, 1),

**'6'**: (1, 2),

**'\*'**: (1, 3),

**'('**: (1, 4),

**'1'**: (2, 0),

**'2'**: (2, 1),

**'3'**: (2, 2),

**'-'**: (2, 3),

**')'**: (2, 4),

**'0'**: (3, 0),

**'00'**: (3, 1),

**'.'**: (3, 2),

**'+'**: (3, 3),

**'='**: (3, 4),

}

# Create the buttons and add them to the grid layout

for btnText, pos in buttons.items():

self.buttons[btnText] = QPushButton(btnText)

self.buttons[btnText].setFixedSize(40, 40)

buttonsLayout.addWidget(self.buttons[btnText], pos[0], pos[1])

# Add buttonsLayout to the general layout

self.generalLayout.addLayout(buttonsLayout)

# change 05 - end

# Client code

def main():

"""Main function."""

# Create an instance of QApplication

pycalc = QApplication(sys.argv)

# Show the calculator's GUI

view = PyCalcUi()

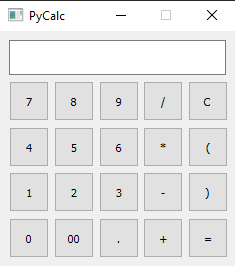
view.show()

# Execute the calculator's main loop

sys.exit(pycalc.exec\_())

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

main()



==================================================

Changes -> 6 to 11

#!/usr/bin/env python3

"""PyCalc is a simple calculator built using Python and PyQt5."""

import sys

# Import QApplication and the required widgets from PyQt5.QtWidgets

from PyQt5.QtWidgets import QApplication

from PyQt5.QtWidgets import QMainWindow

from PyQt5.QtWidgets import QWidget

# change 01 - start

from PyQt5.QtCore import Qt

from PyQt5.QtWidgets import QGridLayout

from PyQt5.QtWidgets import QLineEdit

from PyQt5.QtWidgets import QPushButton

from PyQt5.QtWidgets import QVBoxLayout

# change 01 -end

# change 09

from functools import partial

\_\_version\_\_ = **'0.1'**

\_\_author\_\_ = **'Malith Thiwanka'**

# Create a subclass of QMainWindow to setup the calculator's GUI

class PyCalcUi(QMainWindow):

"""PyCalc's View (GUI)."""

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

"""View initializer."""

super().\_\_init\_\_()

# Set some main window's properties

self.setWindowTitle(**'PyCalc'**)

self.setFixedSize(235, 235)

# Set the central widget and the general layout

# change 02

self.generalLayout = QVBoxLayout()

self.\_centralWidget = QWidget(self)

self.setCentralWidget(self.\_centralWidget)

# change 03 stat

self.\_centralWidget.setLayout(self.generalLayout)

# Create the display and the buttons

self.\_createDisplay()

self.\_createButtons()

# change 03 - end

# change 04 - start

def \_createDisplay(self):

"""Create the display."""

# Create the display widget

self.display = QLineEdit()

# Set some display's properties

self.display.setFixedHeight(35)

self.display.setAlignment(Qt.AlignRight)

self.display.setReadOnly(True)

# Add the display to the general layout

self.generalLayout.addWidget(self.display)

# change 04 - end

# change 05 - start

def \_createButtons(self):

"""Create the buttons."""

self.buttons = {}

buttonsLayout = QGridLayout()

# Button text | position on the QGridLayout

buttons = {**'7'**: (0, 0),

**'8'**: (0, 1),

**'9'**: (0, 2),

**'/'**: (0, 3),

**'C'**: (0, 4),

**'4'**: (1, 0),

**'5'**: (1, 1),

**'6'**: (1, 2),

**'\*'**: (1, 3),

**'('**: (1, 4),

**'1'**: (2, 0),

**'2'**: (2, 1),

**'3'**: (2, 2),

**'-'**: (2, 3),

**')'**: (2, 4),

**'0'**: (3, 0),

**'00'**: (3, 1),

**'.'**: (3, 2),

**'+'**: (3, 3),

**'='**: (3, 4),

}

# Create the buttons and add them to the grid layout

for btnText, pos in buttons.items():

self.buttons[btnText] = QPushButton(btnText)

self.buttons[btnText].setFixedSize(40, 40)

buttonsLayout.addWidget(self.buttons[btnText], pos[0], pos[1])

# Add buttonsLayout to the general layout

self.generalLayout.addLayout(buttonsLayout)

# change 05 - end

# change 06 - start

def setDisplayText(self, text):

"""Set display's text."""

self.display.setText(text)

self.display.setFocus()

# change 06 - end

# change 07 -start

def displayText(self):

"""Get display's text."""

return self.display.text()

# change 7 - end

# change 08 - start

def clearDisplay(self):

"""Clear the display."""

self.setDisplayText(**''**)

# change 08 - end

# change 10 - start

class PyCalcCtrl:

"""PyCalc Controller class."""

def \_\_init\_\_(self, view):

"""Controller initializer."""

self.\_view = view

# Connect signals and slots

self.\_connectSignals()

def \_buildExpression(self, sub\_exp):

"""Build expression."""

expression = self.\_view.displayText() + sub\_exp

self.\_view.setDisplayText(expression)

def \_connectSignals(self):

"""Connect signals and slots."""

for btnText, btn in self.\_view.buttons.items():

if btnText not in {**'='**, **'C'**}:

btn.clicked.connect(partial(self.\_buildExpression, btnText))

self.\_view.buttons[**'C'**].clicked.connect(self.\_view.clearDisplay)

# change 10 - end

# Client code

def main():

"""Main function."""

# Create an instance of QApplication

pycalc = QApplication(sys.argv)

# Show the calculator's GUI

view = PyCalcUi()

view.show()

# change 11

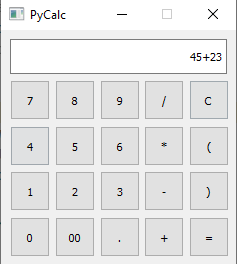
PyCalcCtrl(view=view)

# Execute the calculator's main loop

sys.exit(pycalc.exec\_())

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

main()



====================================================

Changes -> 12 to 16

#!/usr/bin/env python3

"""PyCalc is a simple calculator built using Python and PyQt5."""

import sys

# Import QApplication and the required widgets from PyQt5.QtWidgets

from PyQt5.QtWidgets import QApplication

from PyQt5.QtWidgets import QMainWindow

from PyQt5.QtWidgets import QWidget

# change 01 - start

from PyQt5.QtCore import Qt

from PyQt5.QtWidgets import QGridLayout

from PyQt5.QtWidgets import QLineEdit

from PyQt5.QtWidgets import QPushButton

from PyQt5.QtWidgets import QVBoxLayout

# change 01 -end

# change 09

from functools import partial

\_\_version\_\_ = **'0.1'**

\_\_author\_\_ = **'Malith Thiwanka'**

# change 12

ERROR\_MSG = **'ERROR'**

# change 13 - start

def evaluateExpression(expression):

"""Evaluate an expression."""

try:

result = str(eval(expression, {}, {}))

except Exception:

result = ERROR\_MSG

return result

# change 13 - end

# Create a subclass of QMainWindow to setup the calculator's GUI

class PyCalcUi(QMainWindow):

"""PyCalc's View (GUI)."""

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

"""View initializer."""

super().\_\_init\_\_()

# Set some main window's properties

self.setWindowTitle(**'PyCalc'**)

self.setFixedSize(235, 235)

# Set the central widget and the general layout

# change 02

self.generalLayout = QVBoxLayout()

self.\_centralWidget = QWidget(self)

self.setCentralWidget(self.\_centralWidget)

# change 03 stat

self.\_centralWidget.setLayout(self.generalLayout)

# Create the display and the buttons

self.\_createDisplay()

self.\_createButtons()

# change 03 - end

# change 04 - start

def \_createDisplay(self):

"""Create the display."""

# Create the display widget

self.display = QLineEdit()

# Set some display's properties

self.display.setFixedHeight(35)

self.display.setAlignment(Qt.AlignRight)

self.display.setReadOnly(True)

# Add the display to the general layout

self.generalLayout.addWidget(self.display)

# change 04 - end

# change 05 - start

def \_createButtons(self):

"""Create the buttons."""

self.buttons = {}

buttonsLayout = QGridLayout()

# Button text | position on the QGridLayout

buttons = {**'7'**: (0, 0),

**'8'**: (0, 1),

**'9'**: (0, 2),

**'/'**: (0, 3),

**'C'**: (0, 4),

**'4'**: (1, 0),

**'5'**: (1, 1),

**'6'**: (1, 2),

**'\*'**: (1, 3),

**'('**: (1, 4),

**'1'**: (2, 0),

**'2'**: (2, 1),

**'3'**: (2, 2),

**'-'**: (2, 3),

**')'**: (2, 4),

**'0'**: (3, 0),

**'00'**: (3, 1),

**'.'**: (3, 2),

**'+'**: (3, 3),

**'='**: (3, 4),

}

# Create the buttons and add them to the grid layout

for btnText, pos in buttons.items():

self.buttons[btnText] = QPushButton(btnText)

self.buttons[btnText].setFixedSize(40, 40)

buttonsLayout.addWidget(self.buttons[btnText], pos[0], pos[1])

# Add buttonsLayout to the general layout

self.generalLayout.addLayout(buttonsLayout)

# change 05 - end

# change 06 - start

def setDisplayText(self, text):

"""Set display's text."""

self.display.setText(text)

self.display.setFocus()

# change 06 - end

# change 07 -start

def displayText(self):

"""Get display's text."""

return self.display.text()

# change 7 - end

# change 08 - start

def clearDisplay(self):

"""Clear the display."""

self.setDisplayText(**''**)

# change 08 - end

# change 10 - start

class PyCalcCtrl:

"""PyCalc Controller class."""

# change - Add parameter

def \_\_init\_\_(self, model, view):

"""Controller initializer."""

# change 14

self.\_evaluate = model

self.\_view = view

# Connect signals and slots

self.\_connectSignals()

# change 15 - start

def \_calculateResult(self):

"""Evaluate expressions."""

result = self.\_evaluate(expression=self.\_view.displayText())

self.\_view.setDisplayText(result)

# change 15 - end

def \_buildExpression(self, sub\_exp):

"""Build expression."""

# change 16 - start

if self.\_view.displayText() == ERROR\_MSG:

self.\_view.clearDisplay()

# change 16 - end

expression = self.\_view.displayText() + sub\_exp

self.\_view.setDisplayText(expression)

def \_connectSignals(self):

"""Connect signals and slots."""

for btnText, btn in self.\_view.buttons.items():

if btnText not in {**'='**, **'C'**}:

btn.clicked.connect(partial(self.\_buildExpression, btnText))

# change 15 - start

self.\_view.buttons[**'='**].clicked.connect(self.\_calculateResult)

self.\_view.display.returnPressed.connect(self.\_calculateResult)

# change 15 - end

self.\_view.buttons[**'C'**].clicked.connect(self.\_view.clearDisplay)

# change 10 - end

# Client code

def main():

"""Main function."""

# Create an instance of QApplication

pycalc = QApplication(sys.argv)

# Show the calculator's GUI

view = PyCalcUi()

view.show()

# # change 11

# PyCalcCtrl(view=view)

model = evaluateExpression

PyCalcCtrl(model=model, view=view)

# Execute the calculator's main loop

sys.exit(pycalc.exec\_())

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

main()

