Dialogs

WEEK 6

Instructor: Ph.D. Oleg Tymchuk

Overview

QInputDialog

QFileDialog

QFontDialog

QColorDialog

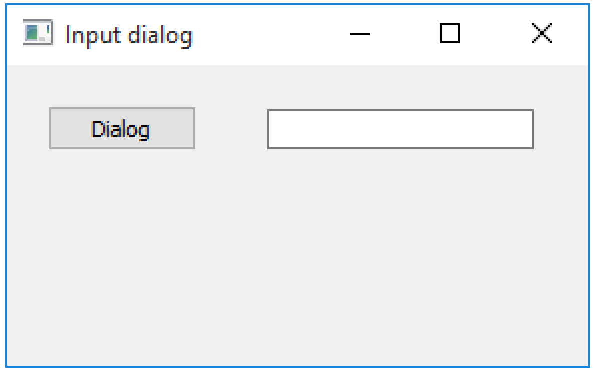
Standard Dialogs

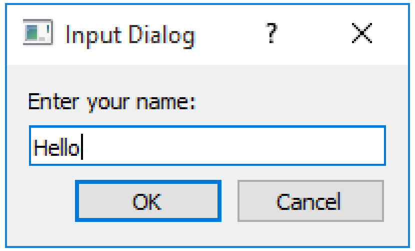
QPageSetupDialog	Configuration dialog for the page-related options on a printer
QPrintDialog	Dialog for specifying the printer's configuration
QPrintPreviewDialog	Dialog for previewing and configuring page layouts for printer output
QColorDialog	Dialog widget for specifying colors
QFileDialog	Dialog that allow users to select files or directories
QFontDialog	Dialog widget for selecting a font
QInputDialog	Simple convenience dialog to get a single value from the user
QMessageBox	Modal dialog for informing the user or for asking the user a question and receiving an answer
QProgressDialog	Feedback on the progress of a slow operation

QInputDialog

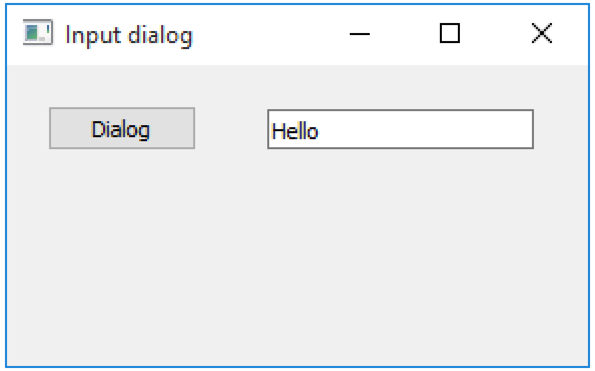
```python
import sys
from PyQt5.QtWidgets import QWidget, QPushButton, QLineEdit,\
    QInputDialog, QApplication

class InputDialogExample(QWidget):
    def __init__(self, parent=None):
        super(InputDialogExample, self).__init__(parent)
        self.btn = QPushButton('Dialog', self)
        self.btn.move(20, 20)
        self.btn.clicked.connect(self.showDialog)
        self.le = QLineEdit(self)
        self.le.move(130, 22)

        self.setGeometry(300, 300, 290, 150)
        self.setWindowTitle('Input dialog')
        self.show()

    def showDialog(self):
        text, ok = QInputDialog.getText(self, 'Input Dialog',
                                        'Enter your name:')
        if ok:
            self.le.setText(str(text))

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = InputDialogExample()
    sys.exit(app.exec())
```

QInputDialog





Enter your name: Hello



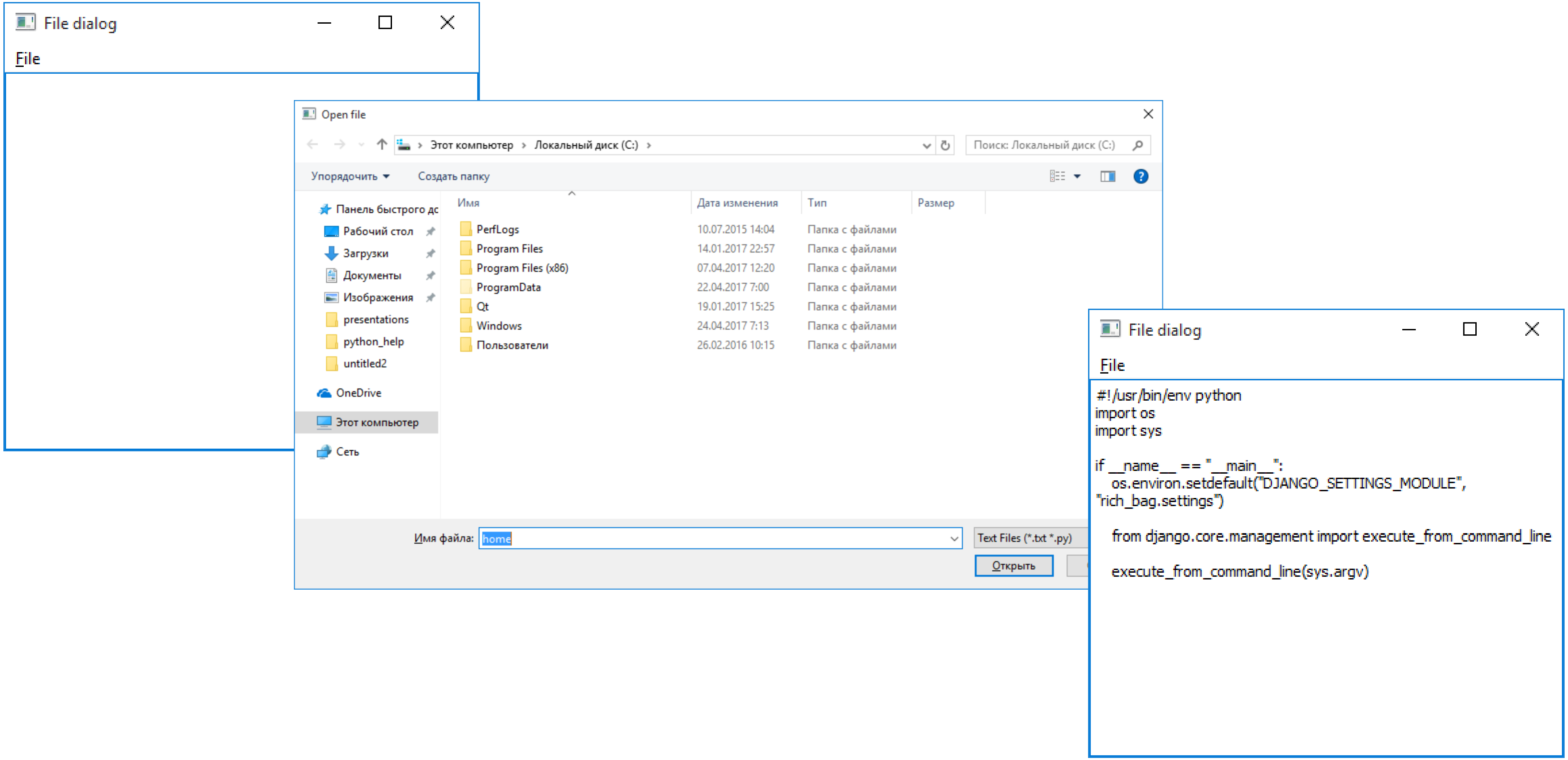
Hello

QFileDialog

```
import sys
from PyQt5.QtWidgets import QMainWindow, QTextEdit,
        QAction, QFileDialog, QApplication

class QFileDialogExample(QMainWindow):

    def __init__(self, parent=None):
        super(QFileDialogExample, self).__init__(parent)
        self.textEdit = QTextEdit()
        self.setCentralWidget(self.textEdit)

        openFile = QAction('Open', self)
        openFile.setShortcut('Ctrl+O')
        openFile.triggered.connect(self.showDialog)

        menu = self.menuBar()
        fileMenu = menu.addMenu('&File')
        fileMenu.addAction(openFile)

        self.setGeometry(300, 300, 350, 300)
        self.setWindowTitle('File dialog')
        self.show()
```

```
def showDialog(self):
        fileName = QFileDialog.getOpenFileName(self,
                 'Open file', '/home',
                 'Text Files (*.txt *.py)')
        if fileName[0]:
            f = open(fileName[0], 'r')
            with f:
                data = f.read()
                self.textEdit.setText(data)


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = QFileDialogExample()
    sys.exit(app.exec())
```

QFileDialog



Имя	Дата изменения	Тип	Размер
PerfLogs	10.07.2015 14:04	Папка с файлами	
Program Files	14.01.2017 22:57	Папка с файлами	
Program Files (x86)	07.04.2017 12:20	Папка с файлами	
ProgramData	22.04.2017 7:00	Папка с файлами	
Qt	19.01.2017 15:25	Папка с файлами	
Windows	24.04.2017 7:13	Папка с файлами	
Пользователи	26.02.2016 10:15	Папка с файлами	

Имя файла: home

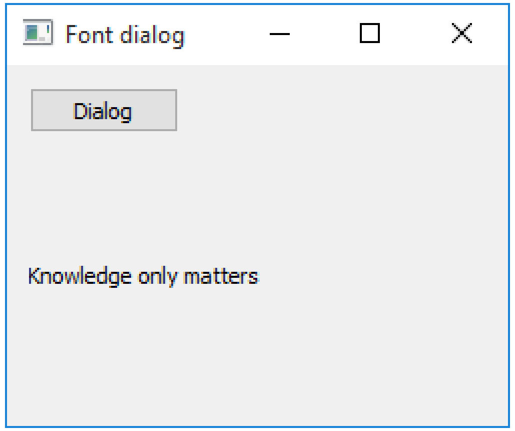
Text Files (*.txt *.py)

```
#!/usr/bin/env python
import os
import sys

if __name__ == "__main__":
    os.environ.setdefault("DJANGO_SETTINGS_MODULE",
"rich_bag.settings")

    from django.core.management import execute_from_command_line

    execute_from_command_line(sys.argv)
```

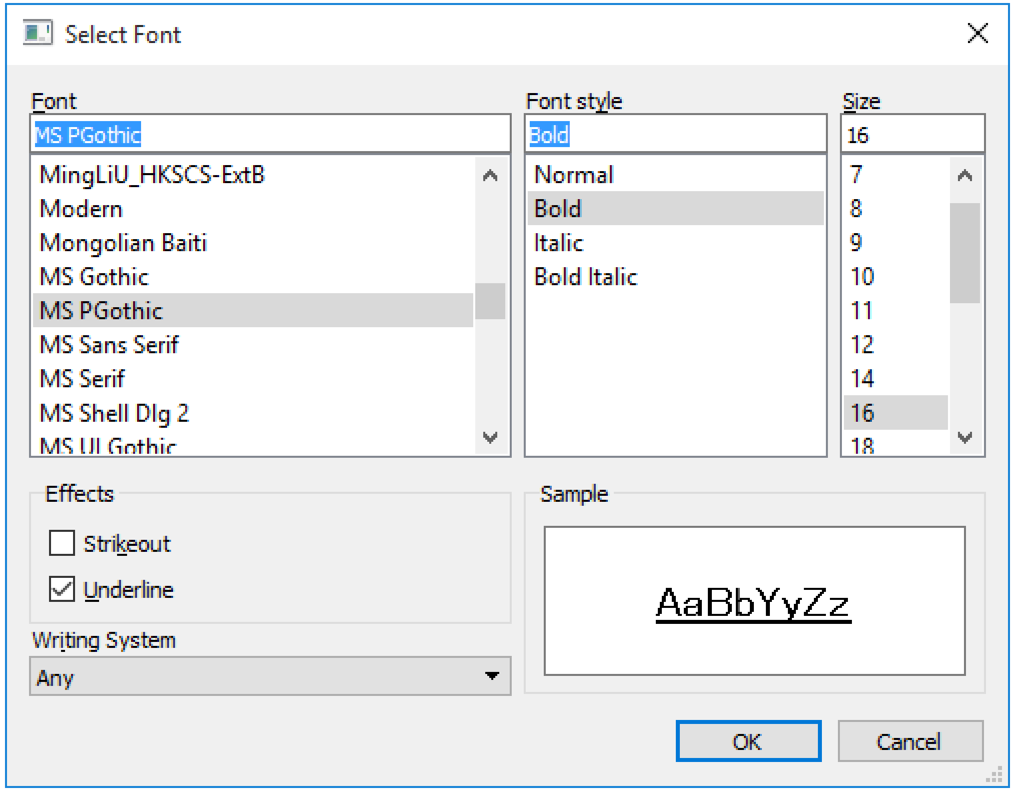
QFontDialog

```python
import sys
from PyQt5.QtWidgets import QWidget, QVBoxLayout,\
    QPushButton, QSizePolicy, QLabel, QFontDialog,\
    QApplication

class QFontDialogExample(QWidget):

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

        vbox = QVBoxLayout()

        btn = QPushButton('Dialog', self)
        btn.setSizePolicy(QSizePolicy.Fixed, QSizePolicy.Fixed)
        btn.move(20, 20)
        btn.clicked.connect(self.showDialog)

        self.lbl = QLabel('Knowledge only matters', self)
        self.lbl.move(130, 20)

        vbox.addWidget(btn)
        vbox.addWidget(self.lbl)
        self.setLayout(vbox)

        self.setGeometry(300, 300, 250, 180)
        self.setWindowTitle('Font dialog')
        self.show()

    def showDialog(self):
        font, ok = QFontDialog.getFont()
        if ok:
            self.lbl.setFont(font)

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = QFontDialogExample()
    sys.exit(app.exec())
```

QFontDialog



Knowledge only matters



Font: MS PGothic

Font style: Bold

Size: 16

Effects

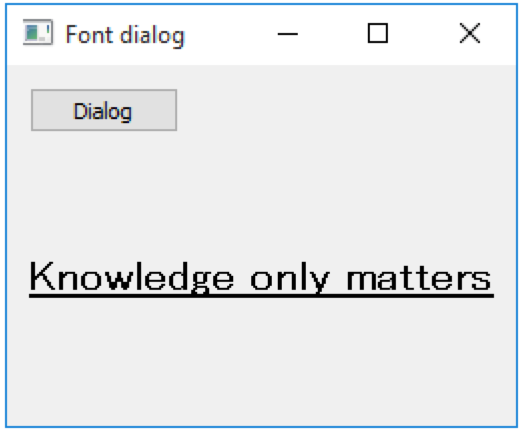
□ Strikeout

☑ Underline

Writing System: Any

Sample

AaBbYyZz



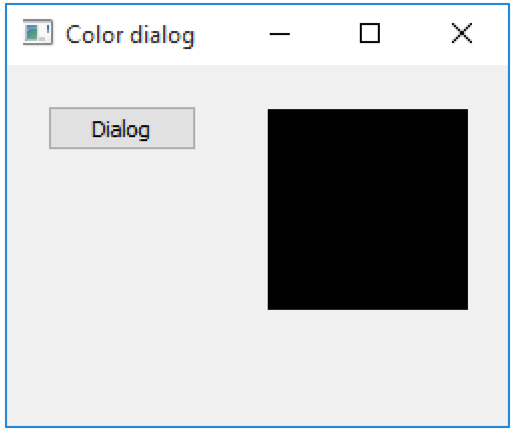
Knowledge only matters

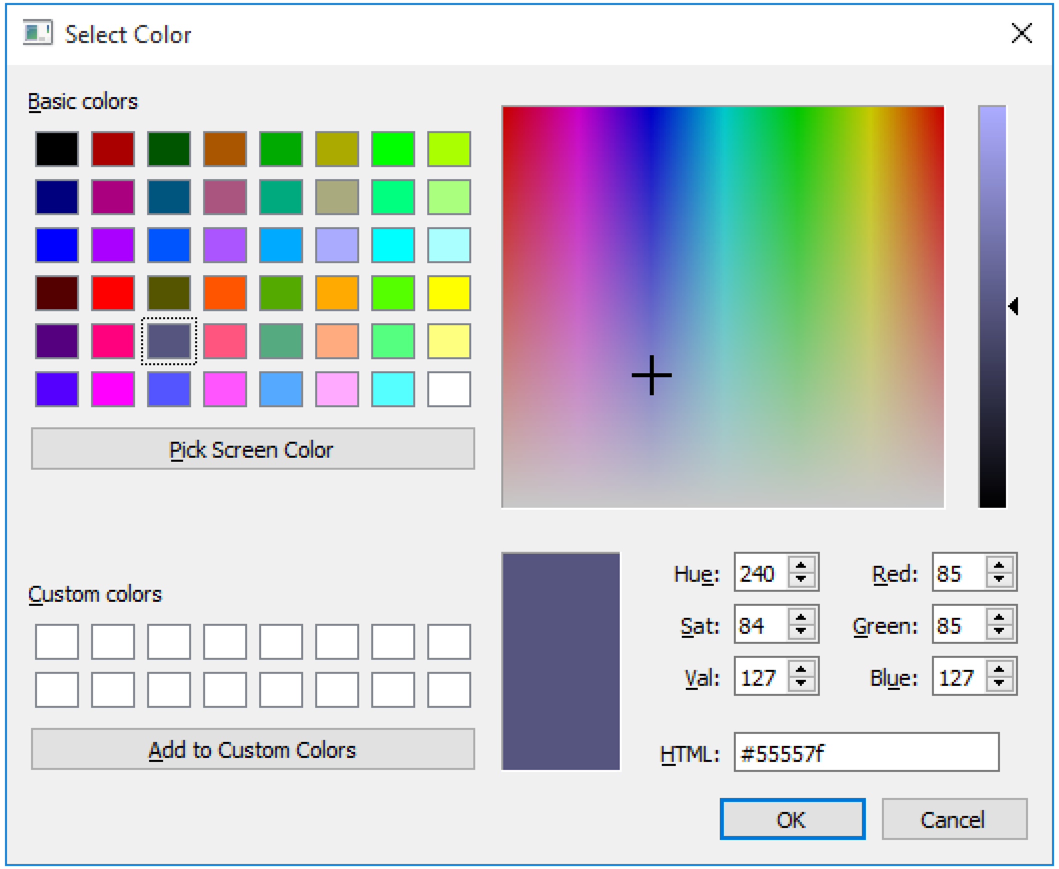
QColorDialog

```python
import sys
from PyQt5.QtWidgets import QWidget, QPushButton,\
    QFrame, QColorDialog, QApplication
from PyQt5.QtGui import QColor

class QColorDialogExample(QWidget):

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

        self.btn = QPushButton('Dialog', self)
        self.btn.move(20, 20)
        self.btn.clicked.connect(self.showDialog)

        self.frm = QFrame(self)
        self.frm.setStyleSheet('QWidget { background-color: %s }'
                               % QColor(0, 0, 0).name())
        self.frm.setGeometry(130, 22, 100, 100)
        self.setGeometry(300, 300, 250, 180)
        self.setWindowTitle('Color dialog')
        self.show()

    def showDialog(self):
        col = QColorDialog.getColor()
        if col.isValid():
            self.frm.setStyleSheet('QWidget { background-color: %s }'
                                   % col.name())
```

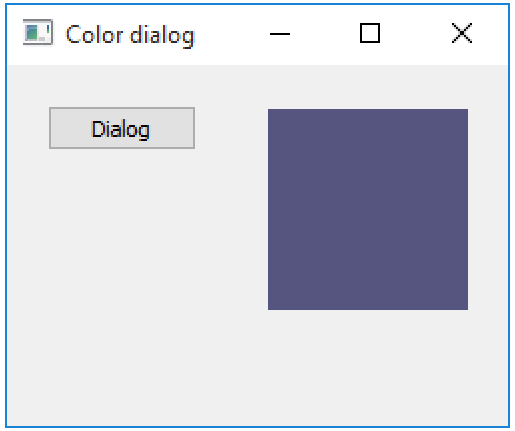
```python
if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = QColorDialogExample()
    sys.exit(app.exec())
```

QColorDialog





Hue: 240
Sat: 84
Val: 127
Red: 85
Green: 85
Blue: 127
HTML: #55557f


Color dialog
Dialog