

PYQT6 Course

2. Working with QPushButton

The push button, or command button, is perhaps the most commonly used widget in any graphical user interface. Push (click) a button to command the computer to perform some action, or to answer a question. Typical buttons are OK, Apply, Cancel, Close, Yes, No and Help.

A command button is rectangular and typically displays a text label describing its action. A shortcut key can be specified by preceding the preferred character with an ampersand in the text. to display a push button in an application, you need to create an instance of the QPushButton class.

There are different methods that you can use in QPushButton.

- `setText()`: This method is used to assign text to the push button
- `setIcon()`: This method is used to assign an icon to the push button
- `setGeometry()`: This method is used for setting the x and y position, also width and height of the button.
- `setMenu()`: This method is used for setting pop menu to the button.

To create a push button in an application, you need to create an instance of the QPushButton class.

```
btn = QPushButton("Click", self)
```

This is used for setting the font of the QPushButton, setFont() method expects a QFont class.

```
btn.setFont(QFont("Times", 14,  
QFont.Weight.ExtraBold))
```

Also using setIcon() you can set an icon to the QPushButton, setIcon() expects QIcon class.

```
btn.setIcon(QIcon("images/python.png"))
```

Using setMenu() we can set a pop menu to the QPushButton object, for that first you need to create QMenu object, QMenu class is related to QtWidgets module, The QMenu class provides a menu widget for use in menu bars, context menus, and other popup menus.

A menu widget is a selection menu. It can be either a pull-down menu in a menu bar or a standalone context menu. Pull-down menus are shown by the menu bar when the user clicks on the respective item or presses the specified shortcut key.

In this line of code first we have created the object of QMenu class, after that we have added some menu items to the menu and at the end by using setMenu() method we have set the popup menu to the button.

```
menu = QMenu()
menu.setFont(QFont("Times", 14,
QFont.Weight.ExtraBold))
menu.setStyleSheet('background-color:green')
menu.addAction("Copy")
menu.addAction("Cut")
menu.addAction("Paste")
btn.setMenu(menu)
```