# PYQT6 Course

## 4. Layout Management QHBoxLayout

In this lesson we are going to start our layout management journey in PyQt6, we have already learned about some basic widgets like QLabel, QPushButton and QLineEdit. When you are developing a GUI application, then layout management is one of the important concepts, luckily in PyQt there are different layout system that you can use, for example we have QHBoxLayout, QVBoxLayout and QGridLayout, in this lesson we want to create an example of QHBoxLayout, QHBoxLayout is related to QtWidgets module and using QHBoxLayout we can align our widgets horizontally, there are some important methods that you can use in QHBoxLayout.

- addWidget(): This method is used for adding widgets to the layout.
- addLayout(): This method is used for adding an specific layout to another layout.
- addSpacing(): This method is used for adding space from the right side.
- addStretch(): This method is used for stretching all the widgets in the horizontal box layout.

For creating of horizontal box layout, first you need to create the object of QHBoxLayout.

```
hbox = QHBoxLayout()
```

When you are working with layout management then you should have some widgets to add that in the layout, in our case we are building a simple layout system and we want to add some QPushButton in the QHBoxLayout, so first you need to create some buttons.

```
btn1 = QPushButton("Click One")
btn2 = QPushButton("Click Two")
btn3 = QPushButton("Click Three")
btn4 = QPushButton("Click Four")
```

For adding the buttons in the QHBoxLayout, we need to use a method from QHBoxLayout that is called addWidget().

```
hbox.addWidget(btn1)
```

One important point that you should take a note, when creating layouts and after adding widgets to the layout, you need to set the main window layout to the specific layout.

```
self.setLayout(hbox)
```