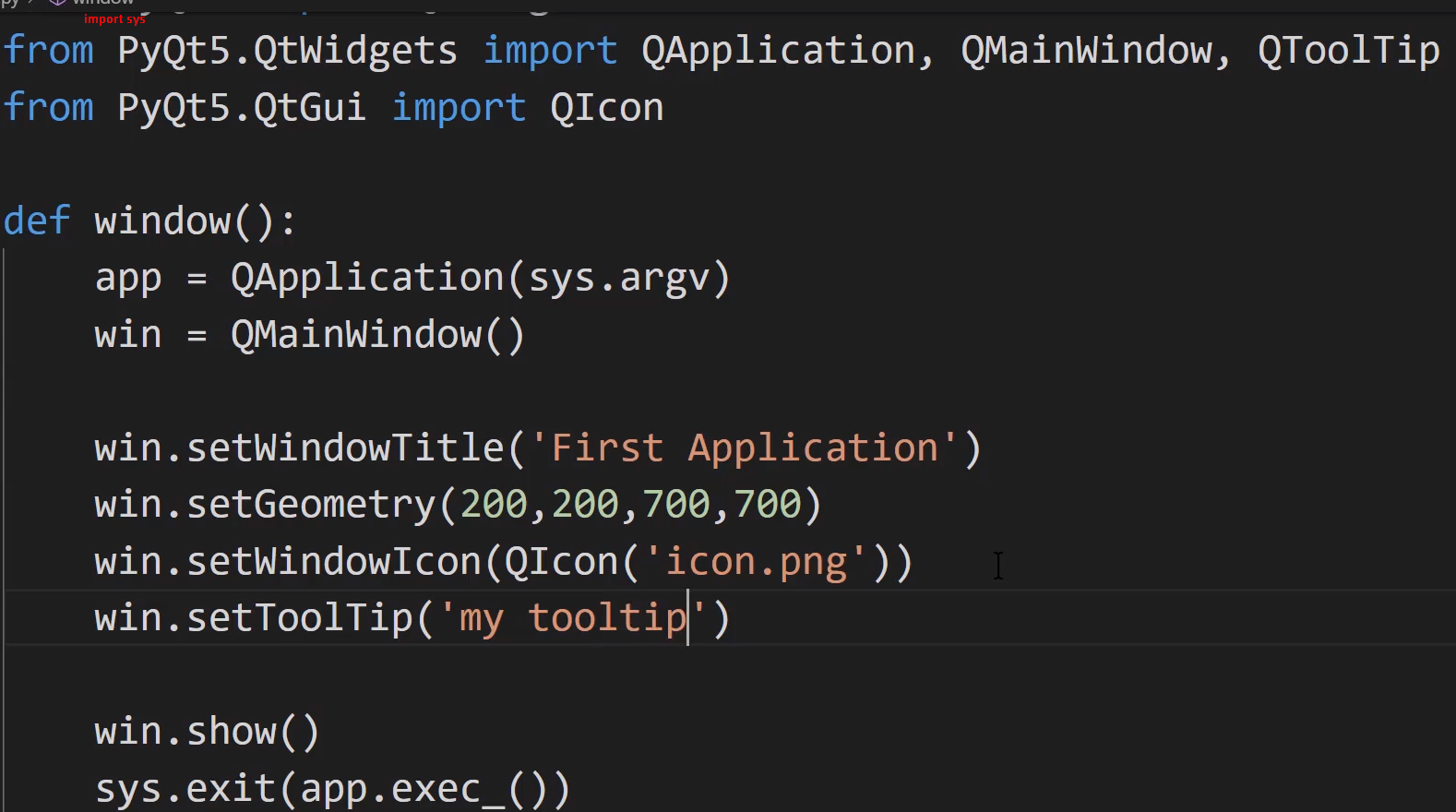
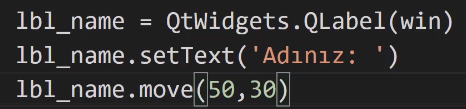
Komut satırından koda argüman göndermek için “import sys” ifadesinin girmemiz gerekir.

[**Kaynak**](https://python-istihza.yazbel.com/standart_moduller/sys.html) **🡪** sys modülü için kaynak.

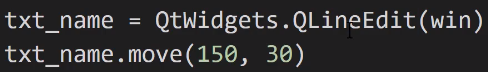


“import sys” komutu ile pencere kapama gibi temel özelliklerin önünü açtık. PyQt5 kütüphanesinden yukarıdaki özellikleri kısaca kullanabilmek için gerekli yapıyı kullanmamızın ardından “window()” adında bir kütüphane oluşturduk. İlk satır ile komut bilgisini uygulamaya aktarmış olduk. “sys.argv” komutu program çalışırken uygulanan parametreleri liste halinde tutmaktadır. İkinci satır ile bir pencere oluşturup sondan önceki “win.show()” satırı ile bu pencerenin arayüz olarak gözükmesini sağladık. Son satırdaki “sys.exit(app.exec\_())” komut ile sisteme bir çıkış yolu ekledik ve bu çıkış yolunun sağ üst köşedeki çarpı olmasını sağladık.

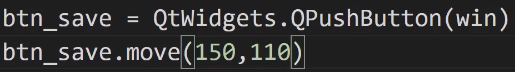
Üçüncü satır ile penceremizin sol üstte görünen başlığını değiştirerek bir alt satırıyla bu pencerenin boyutunu ayarladık (*200, 700 ifadelerini anlayamadım denemek lazım*). Beşinci satır yardımıyla ana başlıkta görülecek olan mini logoya fotoğraf dosyası yükledik. Altıncı ve son satır sayesinde ise fare pencere üzerine geldiğinde “my tooltip” yazısı fare üzerinde görünür oldu.

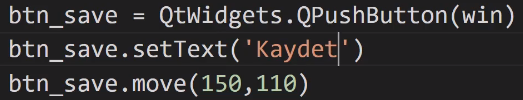


Örnekteki ilk komut ile “win“ penceresi içerisine “Qlabel()” komutu ile bir label koyarak ikinci komut ile bu label’a “Adınız: “ isimli text yerleştirdik. Üçüncü satırda ise bu label’ı yeniden konumlandırdık (konumlandırma yapılmazsa konum (0, 0) olur).



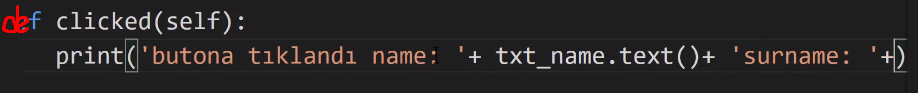
Örnekteki ilk komut sayesinde “QLineEdit()” ile win penceresine içine string yazılabilecek bir alan yerleştirdirerek ikinci satır ile bu alanın konumunu belirledik.







Örnekteki ilk satırda “QPushButton()” ile pencereye bir buton yerleştirerek üçüncü komut ile konumu belirledik. “setText()” komutu ile de buton içerisine “Kaydet” stringini yazdırdık.



Yukarıdaki ilk iki örneğin arasına girmesi gereken “clicked()” fonksiyonu (ilgili pusbutton yapısından önce gelmesi gerekiyor) sayesinde isim ve soyisim bilgisinin girilmesinin ardından sistemde string halinde gözükecek bir metot tanımladık. Üçüncü örnek “.clicked.connect(…)” sayesinde “clicked” metodumuzu butona bağlamış olduk.

“.resize(x, y)” komutuyla belirlenen bileşenin x ve y eksenlerinde boyutları güncellenebilir.

“.setText(“…”)” komutunu kullanarak clicked metodu içerisindeki stringi diğer metotlar yardımıyla belirlenen yere yazdırabiliriz.



Fonksiyonu ile “.sender()” olarak atayacağımız komut ilgili olan butonları sistemin algılamasını ve ona göre tepki vermesini sağlar.

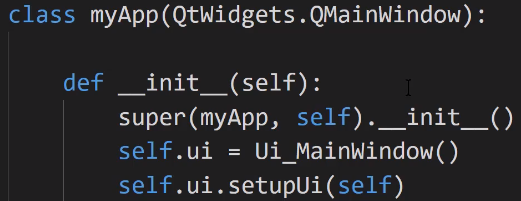
MainWindow içerisine QWidget oluşturabiliriz ve oluşturduğumuz widget mainwindowun bir subclassı haline gelir. mainWindow’da “menubar”’a tıklayarak menü çubuğu oluşturabiliriz.

Qt’deki Object Inspector kısmından bileşenlerin ismini pythona uyarlamak için uygun bir isim vermek önemlidir.

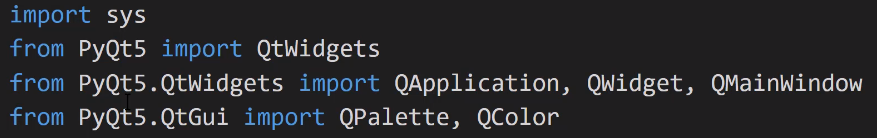
Bu uyarlamayı python diline çevirmek için öncelikle kodu VS Code’da açıp terminal kısmına “pyuic5” (py converter) “dönüştürülecek dosya.ui” -o “dönüştürülecek dosya.py” yazarak yeni bir kod dosyası oluşturulur. Bu oluşturulmuş dosya oluşturduğumuz pencerenin python dilinde yazılmış halidir. Pencere üzerinde yapacağımız herhangi bir değişiklik Qt üzerinde kaydedilip python dosyasını yeniden çalıştırılarak kod güncellemesi yapılır ve değişiklik aktarılır.

Qt kod dosyası üzerinde değişiklik yapmamak önemlidir çünkü yapacağımız her değişiklik, kodu sonraki çalıştırdığımızda güncellenerek silinecektir.

Bir Qt uygulaması için standart başlangıç kodu aşağıda gösterilmiştir:



Layout komutlarını aktifleştirmeden önce aşağıdaki komutları sisteme belirli şekilde entegre ettik:



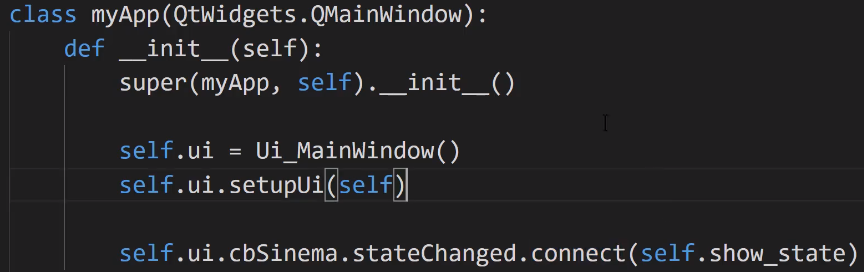
Yukarıdaki komutları öncelikle uygulayarak kaynaktaki bilgiler yardımıyla penceremizde arka plan rengi oluşturma, pencere içerisi renkler ile yatay ve dikey layoutlar oluşturup bu kısımları birbirinden ayrı şekilde kullanma gibi komutları öğrenebiliriz 🡪 [**Kaynak**](https://www.btkakademi.gov.tr/portal/course/player/deliver/s-f-rdan-ileri-seviye-python-programlama-5877)

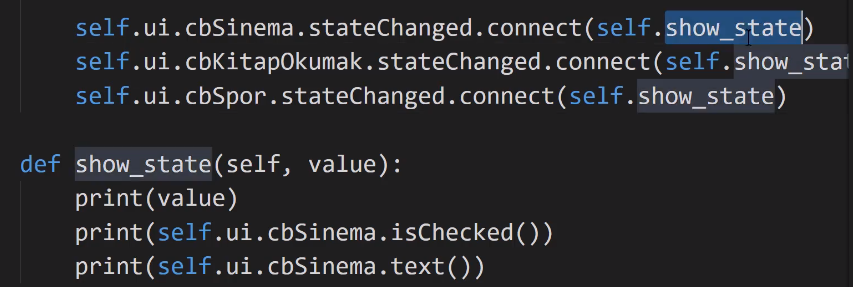
Fakat bu işlemleri Qt arayüzü yardımıyla çok daha kolay ve pratik bir biçimde yapabileceğimizden kodlamanın çok gerekli olduğunu düşünmüyorum.

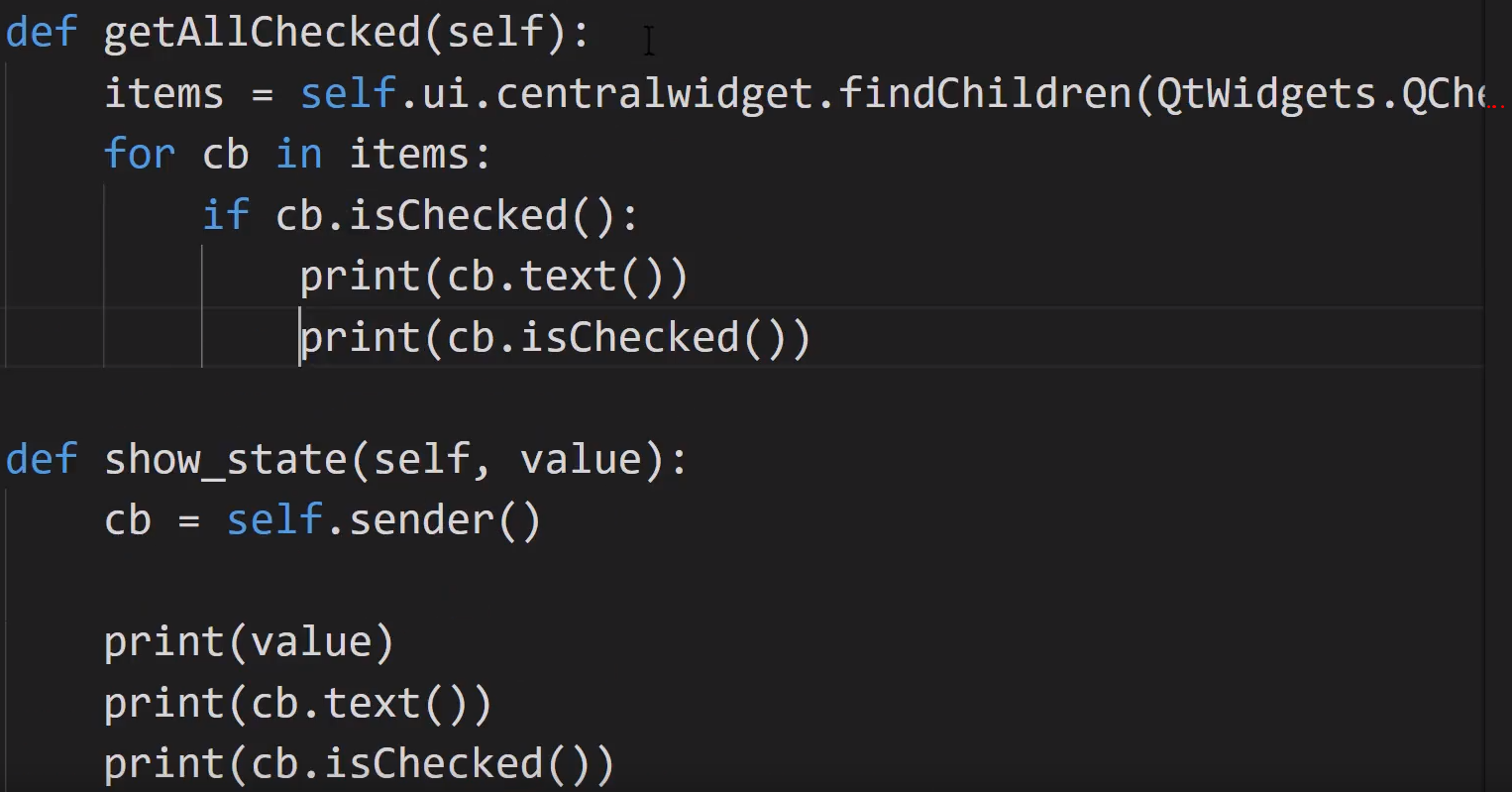
Horizontal ve Vertical Layout bileşenlerini kullanarak yatay ve dikey şekilde aynı çizgide olacak şekilde bileşenleri sıralayabilir, property kısmındaki layout kısmından bileşenlerin konumunu ve boyutunu ayarlayabiliriz. QWidget kısmındaki policy kısmından bileşen boyutuna farklı bir olayla değişiklik yapabiliriz. Spacer kullanarak layout içerisindeki bileşenler arasına ayarlanabilir boşluk ekleyebiliriz.

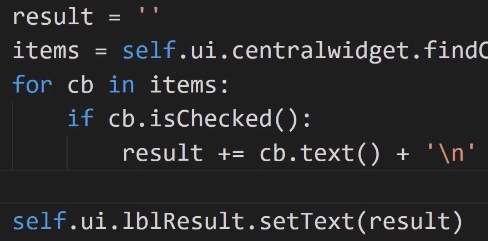
Sağ tık 🡪 Layout diyerek horizontal, vertical veya grid layout’u daha kolay yapabiliriz. Bunu uyguladığımız takdirde main windowun tamamı seçtiğimiz layout türüyle kaplanacaktır (normalde bir kutu oluşur ve bileşenleri onun içine yerleştiririz). Başka bir komut olarak sağ tık 🡪 Form Layout dediğimizde girdiğimiz bileşenler düzenli bir form yapısıyla hizalanacaktır.

checkBox bileşeni ile “state change” olayı gerçekleşir ve seçilen bileşen koda aktarılabilir.









Şeklinde pencereye sırayla nelerin seçildiğini yansıtabiliriz. Üstteki son dört örneğin olduğu kod satırları ile checkBoxlardan ve işaretlediğimiz kutu(yu)(ları) butona basınca bize gösteren bir pencere için kullanılmaktadır.

“stateChanged” komutu sayesinde checkBox’a seçilme fonksiyonu kazandırdık.

Qt içerisinde “radioButton” kullanımına gelecek olursak bu butonlar seçim yapmada kullanılır.

Arayüz içerisindeki “Group Box”’lar birden fazla değeri barındırarak grup içerisinde sunmak için idealdir (genellikle grid formunu içerisine yerleştiririz).



“.findChildren()” komutu ile ilgili groupbox bileşenlerine ulaşabiliriz.

Seçimle alakalı bir uygulama yapacağın zaman kaynaktaki videoyu izleyebilirsin 🡪 [**Kaynak**](https://www.btkakademi.gov.tr/portal/course/player/deliver/s-f-rdan-ileri-seviye-python-programlama-5877)

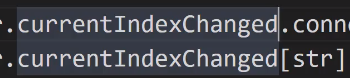
“Combo box” ile aşağıda görülen gibi bir seçim ortamı oluşturabiliriz.

 Bu ortama “Ankara, İstanbul” gibi değerler eklemek içinse aşağıdaki gösterilen şekilde “.addItem(“”) komutunu kullanmamız yeterli olacaktır.



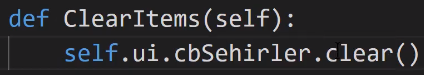
Komutunu kullanarak çoklu ekleme işlemini de uygulayabiliriz.

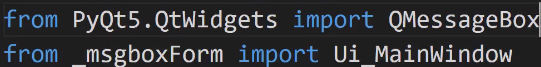
“.currentText()” komutu ile istenilen text bilgisini çekebilir, “.currentIndex” komutu ile de bu text’in indexini alabiliriz.



Üstteki komutlar yardımıyla farklı bir veri seçtiğimiz takdirde değişen verinin ilk kod için indeksi, ikinci kos için string bilgisi getirilir.

“.btnClear” komutu ile butonun belirli verileri sileceği bir kod yazabiliriz.

 “.clear()” komutunu metod içinde kullandık.



“QMessageBox” modülünü aktarıp arayüzümüzü koda çekmemizin ardından kodumuz messageBox işlemlerini görmeye hazır hale gelir. Bu mödül ana pencerenin yanında yeni bir mesaj penceresi açılmasına olanak tanır.

“.btnExit” komutu ile bir butona pencereyi kapama yetkisi verebiliriz. Tabii ki aynı zamanda aynı butona .clicked ve .connect komutlarını da atamamız gerekir.

Metot içerisine yerleştirilecek “.setWindowTitle(“”)” komutu ile string içerisindeki yazıyı sol üstte görüntüleyebiliriz. “setIcon” komutu ile de başlığın yanına bir ikon koyabiliriz. Bu ikonlardan biri “.Question”’dur (.Warning gibi daha fazla ikon internetten bulunabilir) ve soru sorma işlemlerinde kullanılabilir. Pencere içerisine yazı yazdırmak için ise “.setText(“”)” komutunu kullanabiliriz.

“.setStandardButtons(QMessageBox.”…”)” komutunda “…” içerisine “.Ok/.Cancel/.Ignore” ifadelerini yazabiliriz, or operatörü ile (standart “|” kullanılır) bu üç ifadenin ikisini veya üçünü aynı anda kullanmamız mümkün.



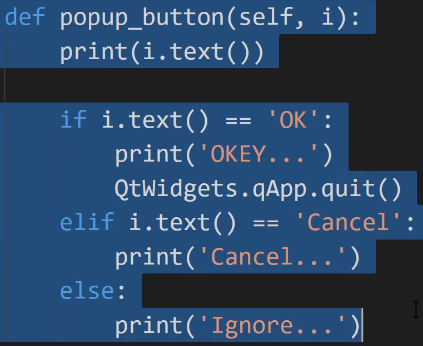
Aşağıdaki default komutu ile de seçili olacak ifadeyi belirleyebiliriz.



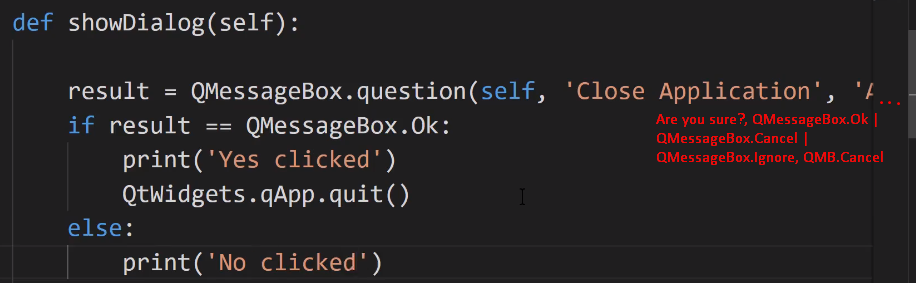
Aşağıdaki komutun kullanımı ise bu mini pencereye ayrı bir buton ekleyecek ve tıklanıldığı takdirde string içerisine yazılanları gösterecek bir alan ortaya çıkaracaktır. 

Aşağıdaki kod bütünüyle mini pencerede tıklanacak butonlara göre sistemin verebileceğifarklı farklı tepkileri ayarladık.





Bunun yerine “.question()” komutu ile hiç uğraşmadan birçok işlemi tek bir satırda halledebiliriz.

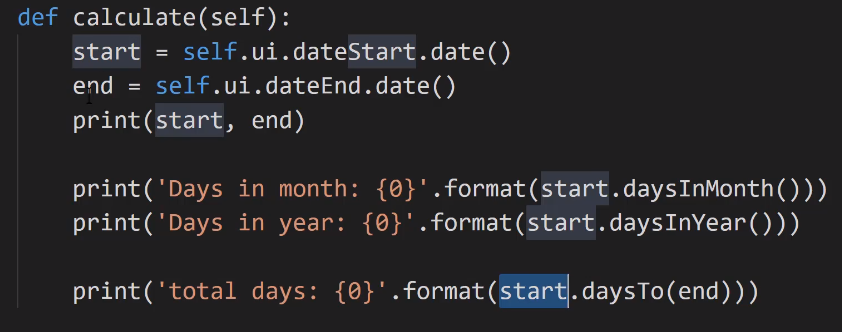


Burada question içerisindeki ilk parametre mini pencere başlığını, ikinci parametre pencere içinde yazılacak stringi, üçüncü yapametre gösterilecek butonları ve son parametre ise default olarak belirlediğimiz (başlangıçta seçili olarak gelen buton) butonu ifade eder.



Yukarıdaki ifade ile Qt için zaman kütüphanesini import edebiliriz.

“.date()” komutu ile Qt penceresinde belirlenmiş zaman ifadesini çekebilir, “param.daysInMonth”, “param.daysInYear” komutlarıyla ay ve yıl içerisinde kaç gün olduğunu ifade edebiliriz.

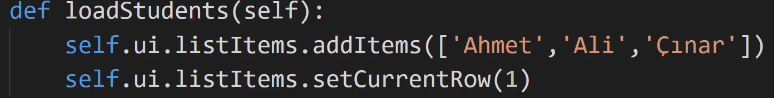


“param1.daysTo(param2)” komutu ile belirlenen iki tarih arasında kaç gün fark olduğunu öğrenebiliriz. “.currentDate()” komutu ile belirlenen tarih ile güncel tarih arasındaki zaman farkını gün cinsinden öğrenebiliriz (bir parametreyi bu komuta eşitleyerek örnekteki son satır gibi uygulamamız gerekir).

Hep gün üzerinden gittiğimiz bu zaman kavramı saat gibi zaman dilimleri üzerinden de uygulanabilir durumdadır. Komutları öğrenmek için araştırılabilir.



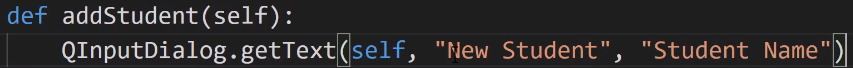
Yukarıdaki ifadeyi import ederek pencere içerisinde liste oluşturabiliriz. “.addItems([“,”]) komutu ile liste içerisine string yazdırabiliriz. bu stringler “List Widget” bölümünde gösterilir.

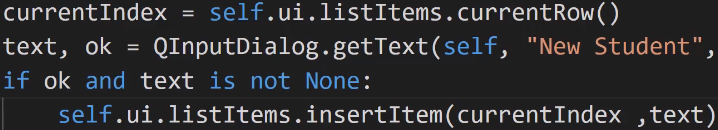


“.setCurrentRow(index)” metodu ile başlangıçta seçili olacak indexi belirleyebiliriz.

Şu anda yaptığımız uygulama, sisteme öğrenci verileri ekleyip bu verileri manipüle edebilen bir özelliğe sahiptir. Uygulamaya veri eklemek istersek uygulamamız gereken bir kütüphane bulunuyor:

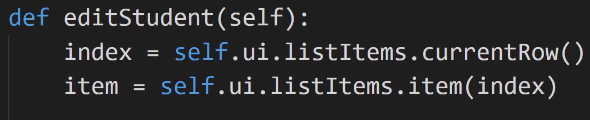


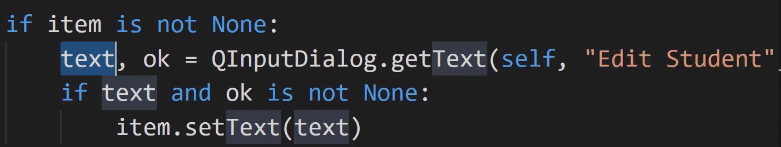




“.getText()” komutu ile açılacak yeni pencereye string yerleştirebiliriz. İlk parametre pencerenin başlığını ifade ediyorken ikinci parametre pencere içerisindeki yazıyı belirtir.

Eğer “ok” ve “text” komutları none olarak dönmezse “insertItem()” komutu “currentIndex”’in text’ini yazılan string ile değiştirir.

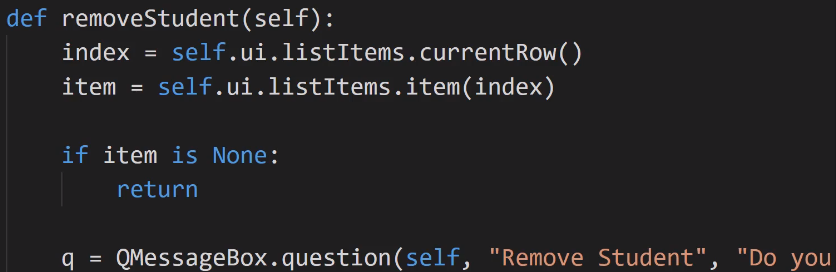




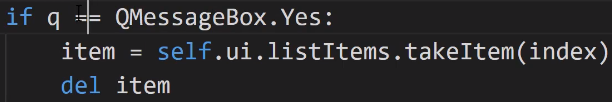


Yukarıdaki örnekte öncelikle verinin sırasını belirledik ve “item“ değişkenine seçtiğimiz elemanı alabilmek için “.item()” komutunu atadık. Eğer seçtiğimiz eleman none değilse öğrenci verisi düzenleme penceresi açılacak, sonrasında girilen string ve basılan “ok” butonu sisteme none değer göndermezse pencere içerisine yazılan yeni veri güncellenecektir.

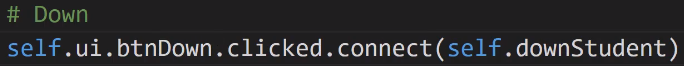
“.remove()” komutunu kullanabilmek için öncelikle QMessageBox modülünü importladık.

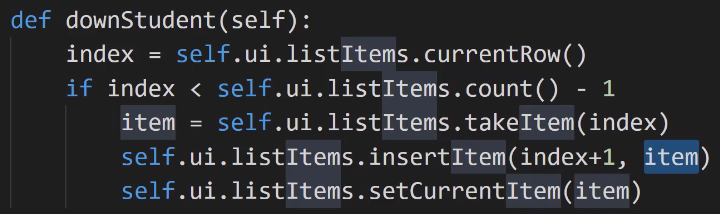




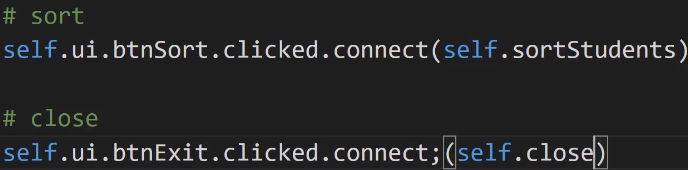


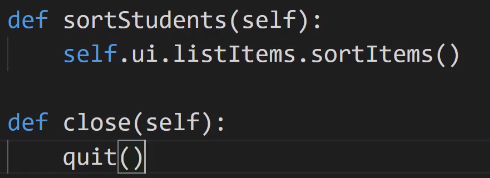
“del item” komutuyla seçilen veriyi rahatlıkla silebiliriz.





Yukarıdaki kod ile pencerede oluşturulan down butonu sayesinde bir verinin indexini bir üst değere taşımayı başardık (aynısı ufak farklılıklarla (daha basit şekilde) up butonu için uygulanabilir).

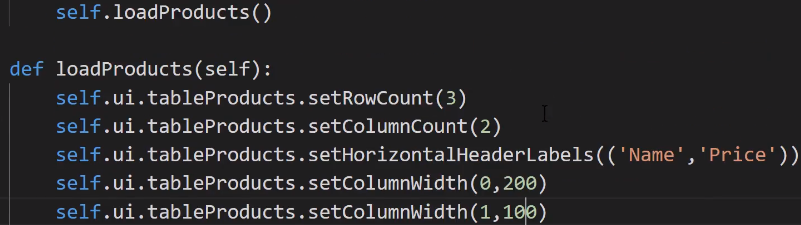


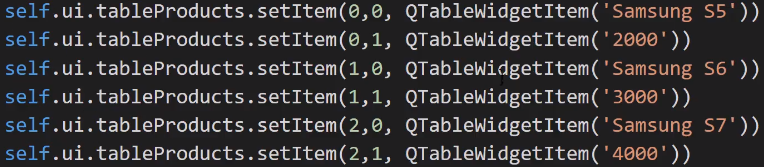


Yukaridaki “sortStudents()” metodu ile “.sortItems()” komutunu kullanarak verilerimizi alfabetik veya sayısal şekilde sıralayabiliriz.

“quit()” komutu ile uygulama penceremizi kapatabiliriz.





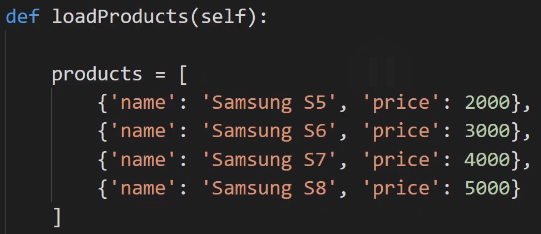


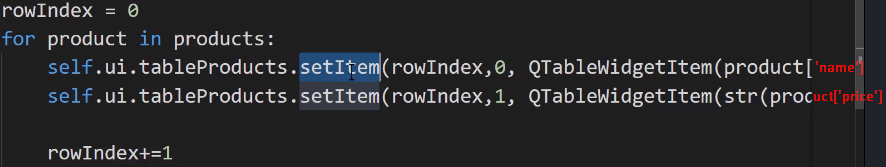
“.setRowCount()” ile parantez içine ifade edeceğimiz numarayla kaç adet sıra oluşturacağımızı belirtiriz, aynı durum “.setColumnCount()” komutu için sütun sayısı için geçerlidir. “.setHorizontalHeaderLabels()” komutu ile sıra başlıktaki alanların isimlerini belirleyebiliriz ( sütunlar için “.setVerticalHeaderLabels()” kullanılır).

“.setColumnWidth(x, y)” komutu ile bir sütunun en ve boy ayarını yapabiliriz (“.setRowWidth(x, y)” komutu ise sıra için kullanılır).

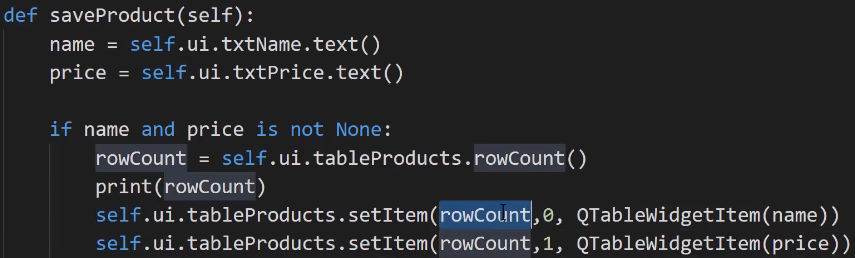
“.setItem(indexrow, indexcol, QTableWidgetItem(“”)” komutu ile belirlediğimiz alana parantez içerisindeki stringi yazdırabiliriz.

Bu uygulamayı dinamik şekilde yapmak için aşağıdaki kod örneğini kullanabiliriz:



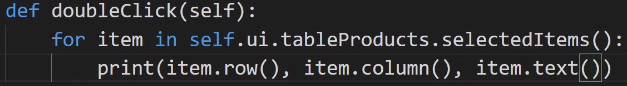






Pencereden aldığımız isim ve fiyat etiketlerini “name” ve “price” değişkenlerine entegre etmemizin ardından bu değişkenler eğer none değeri almazsa listemizin yeni bir satırına bu verilerin girileceği bir metot kurduk. Metodu kurarken if statement içerisindeki ilk komut ile sıra indexini aldık. “.setItem()” komutunu kullanarak belirli sıra ve sütundaki alanlara pencerede belirtilen string değerlerini girdik.





“.doubleClicked” komutunu kullanarak oluşturduğumuz “doubleClick()” metodu ile “.selectedItems()” komutunu kullanarak matriste seçilen hücrenin sıra, sütun ve string değerini printleyen bir kod yazmış olduk.

*24.6/7/8/9 konularına bakmakta fayda var özellikle 24.9*