# Begriffe und Grundlagen bei Qt

Siehe auch folgenden Link: <https://de.wikipedia.org/wiki/Steuerelement>

<https://en.wikipedia.org/wiki/Widget_(GUI)>

<https://en.wikipedia.org/wiki/List_of_graphical_user_interface_elements>

|  |  |
| --- | --- |
| Window | Ein Fenster welches mit den typischen Menüs ergänzt werden kann |
| Widget | Kommt aus der Unix Terminologie und ist ein Visuelles Element in ein GUI. Ist vergleichbar mit „Control“ und „Container“ aus dem Windows Bereich. Widget setzt sich aus dem Wort **Wi**ndows und Ga**dget** zusammen.  Buttons, Menus, Scroll Bars und Rahmen sind ebenfalls Widgets, diese bezeichnet man als sogenannte Item-Widgets. Ein Träger-Widget kann auch andere Widgets aufnehmen. |
| Signal | Wir von Qt zur Verfügung gestellt um Aktionen innerhalb des GUI abzufangen. Folgende Beispiele gibt es dafür:   * Anklicken eines Widgets * Drücken mit Maus innerhalb eines Widgets * Loslassen der Maus innerhalb eines Widgets * Wert hat in einem Eingabefeld geändert * usw. |
| Slot | Ist eine auszuführende Funktion um eine Signal aufzufangen und den enthaltenen Code auszuführen. Qt stellt standardisierte Funktionen Slot’s zur Verfügung oder aber der Programmierer erstellt sich seine eigenen Slot’s mit dem entsprechenden Code. |
| UI / GUI | User Interface (alt Graphic User Interface), Bedienbare grafische Oberfläche des Programmes. |
| Konstruktor | Wird nur einmal ausgeführt sobald das Programm gestartet wird. |
| Destruktor | Wird einmal ausgeführt sobald das Programm beendet wird. |

# Quelldateien und Arbeitsweise

mainwindow.h

#include <iostream>

using namespace std;

class Test{

public:

void input();

private:

int \_a;

int \_b;

};

mainwindow.cpp

(Implementierungsdatei)

#include “header.h”

void test::input(){

…

}

main.cpp

#include “mainwindow.h”

int main(){

…

mainwindow object;

Object.input();

…

return (0);

}

mainwindow.ui

(Userinterface Datei)

#include <QPushButton>

#include <QLineEdit>

#include <QLabel>

….

Programm "mainwindow"

# Variablen erstellen

Um Speicherplatz bei Betriebssystem zu reservieren muss eine Variable vom benötigten Datentyp erstellt werden.

Die Datentypen:

|  |  |  |  |
| --- | --- | --- | --- |
| **Befehl** | **Namen** | **Beispiel** | **Beschreibung** |
| QString | String | “Beispiel 1“ | Zeichenkette |
| int | Integer | 5 | ganze Zahlen |
| double | Double | 5.5 | reelle Zahlen |
| bool | Boolean | false / true | Wahrheitstabelle 0 (false) oder 1 (true) |

Beispiele Variablen erstellen / deklarieren (Speicherplatz reservieren):

QString myString;

int myInt;

double myDouble;

bool myBool;

Beispiel Variablen den Startwert setzen / initialisieren (Wert setzen):

myString = “Hello World”;

myInt = 25;

myDouble = 5.51;

myBool = true;

Deklarieren und Initialisieren gleichzeitig:

QString myString = “Hello World”;

int myInt = 25;

double myDouble = 5.51;

bool myBool = true;

# Bit Operationen

Regeln: AND vor OR

NICHT = !

UND = &&

ODER = | |

# Vergleichsoperationen

== gleich

!= nicht gleich / ungleich

< kleiner als

> grösser als

!< nicht kleiner als

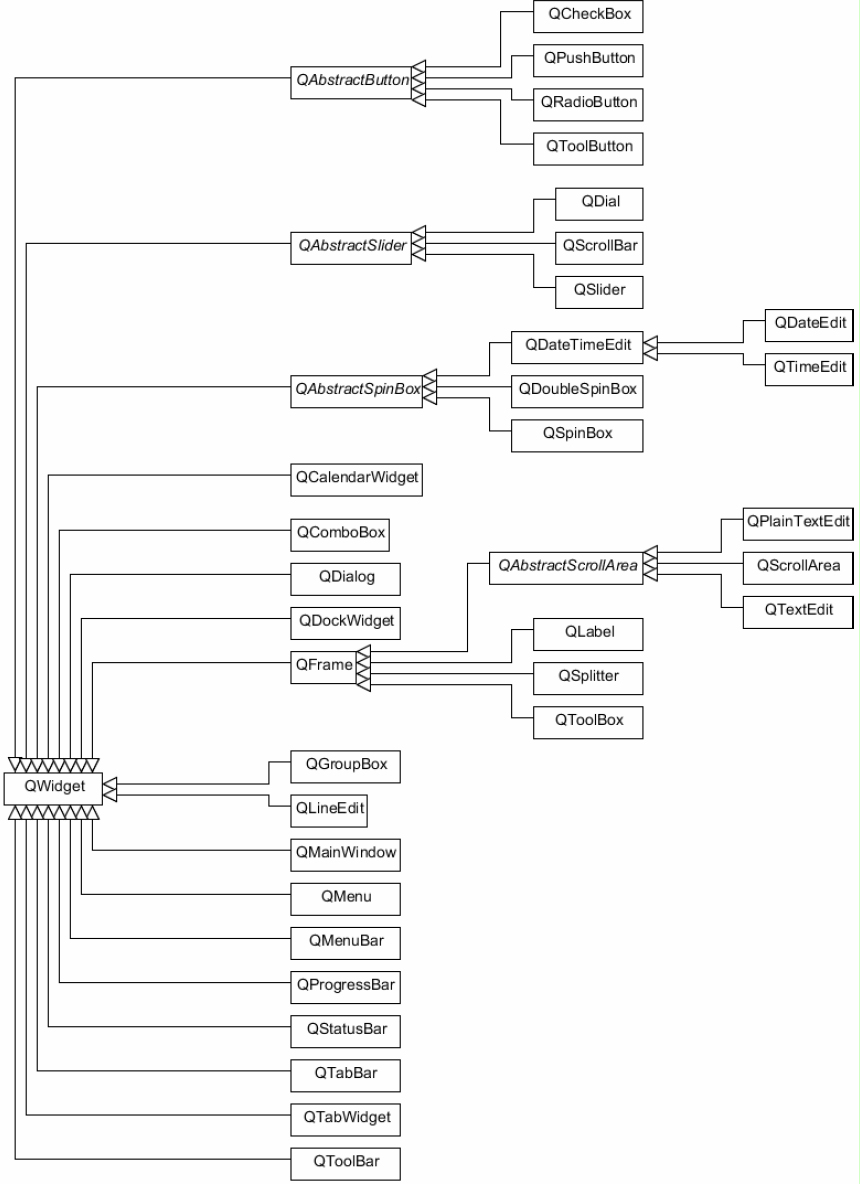
!> nicht grösser als

# Item Widgets

Die wichtigsten Widgets:

|  |  |  |
| --- | --- | --- |
| Aussehen | Namen | Beschreibung |
|  | TextLabel | Beschriftung in einem UI |
|  | LineEdit | Eingabefeld einzeilig |
|  | TextEdit | Eingabefeld mehrzeilig |
|  | PushButton | Druckknopf |
|  | RadioButton | Auswahlknopf, max. einer pro Widget kann „true“ sein |
|  | CheckBox | Auswahlknopf, eine beliebige Anzahl pro Widget kann „true“ sein |
|  | ProgressBar | Zustandsbalken |
|  | ComboBox | Auswahlbox |

Detaillierte Übersicht:



QTableView

QTableWidget

Spreadsheet

QTableWidgetItem

Cell

# WENN Abfrage

|  |  |
| --- | --- |
| **Code** | **Beschreibung** |
| if (x < y) {  ...  Mein gewünschter Code  …  } | if (Bedingung mit Resultat "true" oder "false" ) {  } |
| qDebug() << "ausgeben: " << x; | In Header Datei folgendes eingeben:  #include <QDebug>  Anwendung in cpp Datei: |
| /\*  …  …  …  /\* | Alles was zwischen den Zeichen steht ist Kommentar und wird nicht als Code angeschaut. |
| // … | Alles hinter denn Zeichen auf der Zeile steht ist Kommentar |
|  |  |