# Alkalmi Python, 2. rész

PCLinuxOS Magazine - 2019. március

Írta: Peter Kelly (critter)

#### Egy valós alkalmazás

Most már működik a sablonunk, elkezdhetünk valóban hasznos dolgokat csinálni. Az első alkalmazás a sablonunkból nagyon egyszerűen elkészíthető és oly kevés python nyelvi ismeretet igényel, ahol most valóban tartunk.

Néha, amikor tudod, hogy mit szeretnél mondani, de nem jut eszedbe a kifejezésére alkalmas szó, szinonimaszótárhoz kell fordulni. A szinonimaszótár felsorolja a hasonló, hasonlóan írott és a hasonló jelentésű szavakat. Valójában az összes, általad a szinonimaszótárnak megadott alapján ajánlható szót.

A tárolóban találsz egy "aiksaurus" nevű alkalmazást, ami egy parancssori szinonimaszótár-program. Jó

register

tabulation

stint, straiten

Results:

Enter the search word

to, square, stand together, tally

--- tabulation -----

agree, answer, atone, chime, coincide, concur, conform, cooperate, correspond, defer to, dovetail, go together, go with, hang together, harmonize, hold together, interlock, intersect, jibe, lock, match, overlap, parallel, register, respond

care for, conserve, copyright, guard, keep, keep alive, keep up, maintain, patent, preserve, protect, register, save, support, sustain, tend, uphold

booking, chart, dressing, employment, enlistment, enrollment, entering, entry, filing, impanelment, inscription,

insertion, itemization, listing, logging, matriculation, posting, record keeping, recording, register, registration, registry,

calculator, clerk, comptroller, controller, CPA, engraver, filing clerk, librarian, marker, notary, recorder, register, registrar, scorer, scribe, scrivener, stenographer, stonecutter,

Esc - Ouit

Reset

alany számunkra, 🎫 hogy a sablonunkból grafikus felhasználói felületet készítsünk hozzá. Telepítsd azt és az aiksaurus-data függőséget. Az első verzió úgy néz ki, ahogy a képen látható, de a végleges verzióhoz némileg módosítunk a kinézetén kiegészítjük és némi kód-dal. az alkalmazás működésének javítása érdekében.

A dizájn

Készíts egy thesaurus névű könyvtárat és az alábbiakat másold oda, illetve nevezd át a mutatott módon:

qt5\_template.py ==> thesaurus.py
template.ui ==> thesaurus.ui
update\_res.sh ==> tartsd meg a nevét

Egy megfelelő ikont is rakj ide (a /usr/share/iconsban keress a rendszeredre telepített ikonok között).

Szerkeszd az update\_res.sh fájlt, lecserélve a két "template" szót "thesaurus"-ra valahogy így:

#!/usr/bin/env bash
pyuic5 thesaurus.ui > thesaurus\_ui.py

Indítsd a Qt designer-t és nyisd meg a thesaurus.ui-t. Jobb kattintás a háttéren és válaszd a kinézetet –

break lavout.	A Form (Preview) Ot Designer
Húzással nagyítsd	GroupBox
ki az űrlapot,	TextLabel
valahogy olyanra,	
mint a képen	TextLabel E7 - Clear
látható. Méretét	
később állítjuk be.	
Valójában kézzel	
adjuk meg az	
összes méret- és	
helyzetadatot az	
objektumvizsgálób	
an.	
Adj hozzá egy	
groupbox-ot, egy	
lineedit-et, 2 label-t	
és egy horizontal	
line-t. Ugy rendezd	PyQt5 Template Esc - Quit
el, ahogy a képen	

jobbra látszik és a groupbox-on jobb kattintással válaszd "send to back"-et (háttérbe küldést).

Az Objektumvizsgáló ablaknak a képen látható módon kell kinéznie:

Object Inspector				
Ob	ject			Class
-	-	Fo	rm	QWidget
	*	0	groupBox	QGroupBox
			clearButton	QPushButton
			label	QLabel
			label 2	QLabel
			label 3	QLabel
			line	Line
			lineEdit	QLineEdit
			quitButton	QPushButton
			textEdit	QTextEdit

Most a property editor-ban (tulajdonságok) keresd meg a formát megadó 10 tételt és a következő értékeket állítsd be

Form: width height	440 654			
Put the same <b>maximum size</b> .	values in	minimum	size	and
Horizontal policy vertical policy	set to set to	fixed fixed		
This will stop our remove the need f Window title Window icon Stylesheet	form from for a layout. Thesau Select t backgre	being resiz irus the icon file y ound-color: #	ed an you ch #35383	d so ose 36
Groupbox: x y width height	10 10 420 634			

## Alkalmi Python, 2. rész

Stylesheet title	background-color: #e9eefd clear this field
clearButton: object name x y width height text	reset_button 260 80 150 30 Reset
label2: x y width height Font point size text	20 2 170 20 11 Enter the search word
label3: x y width height Font point size text	10 180 170 24 18 Results:
line: x y width height	10 70 400 3
lineEdit: Objectname x y width height Font point size Stylesheet	lineinput 10 30 400 30 14 background-color: white
quitButton: object name	exit_button

x y width height text	130 583 160 30 Esc – Exit
textedit: object name x y width height Stylesheet frame shape frame shadow readOnly	resultslist 10 120 400 440 background-color: white styled panel sunken check this box
label: Pixmap x y width height	Select the icon file you chose 20 575 48 48
Az űrlapnak most már hogy megfeleljen.	az első képen láthatónak kell,
Ha már jobban ism könnyebben fog men javaslat, kedvedre vá hogy a lineedit, a text objects (objektum) nev python-kódban ezekkel	nered a designer-t, sokkal ni. Az értékek zöme csak Itoztathatsz, de figyelj arra, edit és a két gomb (buttons) re pontos legyen. A végleges fogunk hivatkozni rájuk.
Mentsd az űrlapot és használva készítsd el a	az update_res.sh szkriptet python-fájlt.
A végrehajtható alkalr	nazás elkészítése.
Az aiksaurus parancss kell írni az aiksaurus s	sori eszköz működéséhez be szót, amit a keresett kifejezés

követ. A "register"-t használva így néz ki a példa:

#### aiksaurus register

Egyszerű. De mi ezt egy grafikus ablakban szeretnénk csinálni. Íme a kód – ez alkalommal sorszámok és változtatásokat nincsenek а kiemeltem:

#!/usr/bin/env python3

import sys from PyQt5.QtCore import \* from PyQt5.QtWidgets import \* import subprocess import thesaurus\_ui

class Thesaurus(QWidget, thesaurus\_ui.Ui\_Form): def init (self): super(self.\_\_class\_\_, self).\_\_init\_\_() self.setupUi(self) self.textinput.setFocus() self.textinput.setToolTip( 'Press the return key to displaytheesult') self.clearButton.clicked.connect(self.reset) self.quitButton.clicked.connect(self.exitApplication) self.textinput.returnPressed.connect(self.find it)

def keyPressEvent(self, e): if e.key() == Qt.Key\_Escape: self.exitApplication()

def find\_it(self): self.resultslist.clear() self.target = self.textinput.text() cmd = "aiksaurus " + self.target result = subprocess.check\_output(cmd,shell=True).decode('utf-8') self.resultslist.setText(result)

def reset(self): self.resultslist.clear() self.textinput.clear() self.textinput.setFocus()

def exitApplication(self):

self.close() sys.exit()

if \_\_name\_\_ == '\_\_main\_\_': app = QApplication(sys.argv) form = Thesaurus() form.show() app.exec\_()

(A szerkesztő megjegyzése: a "def find(itself)" résznél a "result =" kezdetű sor egyben van. A következő sor a "self.resultlis" kezdetű.)

Ahogy láthatod, nagyon keveset változtattam, de mégis teljesen más alkalmazás lett.

Egy nagyon fontos kifejezéssel egészítettem ki. Behív egy alárendelt folyamat-modult a szabványos könyvtárból. Az alfolyamat-modul a python része, olyan a kódot tartalmaz, ami a folyamat számára minden esetben terminál parancssorában történő végrehajtását szimulál.

A sablonban az összes "template" hivatkozást "thesaurus"-ra (vagy Thesaurus-ra) cseréltük.

A textedit widget "resultlist" nevet kapott.

A most hozzáadott lineedit widget kapja meg a textedit helyett a fókuszt és azt "textinput"-ra neveztük át, hogy szerepe még világosabb legyen. Hozzáadtam még egy eszközsúgót a textinput-hoz, ami eligazítja a felhasználót. A widget még érzékeli a Return gomb lenyomását is, amivel az alkalmazás kapcsolódik egy find\_it nevű új eljáráshoz.

A clearButton (törlés) a reset nevű eljáráshoz kapcsolódik, ami töröl mindent és a fókuszt visszaadja a textinput.nak.

A find\_it eljárás új. Törli a resultlist nevű texedit widget összes korábbi eredményét. A texinputbox (szövegbevitel) tartalmát "target"-re nevezi át (a "self"-re majd visszatérünk) és készít egy terminál típusú parancsot, amit az aiksaurus vár, összerakva az "aiksurus"-t egy szóközt és a "target"-ünket.

A munkát a következő sor végzi el. Megnézi az összeállított parancs végrehajtásának eredményét egy új folyamatban, ahogy azt a bash is tenné és egy result-nak nevezett változóban tárolja el. A "shell=True" kell az alfolyamatnak, amikor a paramétereket a parancsnak átadja és a decode bit is mindig kell a végére. Az Utf-8 kódolás hatalmas téma, amivel mindenkinek tisztában kell lennie. Ha kíváncsi vagy, akkor erre a legjobb magyarázatot én a http://kunststube.net/encoding/ alatt találtam.

Végül, ebben az eljárásban a kapott szöveget a resultlist txtboxban helyezi el. És ennyi. Készen van. A futó alkalmazásnál vidd az egérmutatót a bemeneti mező fölé az eszközsúgó megjelenítéséhez.

#### A megjelenés

Az alkalmazás megjelenése most még elég egyszerű. Én szeretem, ha az alkalmazásaim kinézete és érzete hasonló. Jelentősen fejleszthető rajtuk stíluslapok használatával, a weblapon alkalmazottakhoz hasonló módon. Stíluslapokkal (a property editorban már használtunk egyes widget-ek színének beállítására) a gombok sarkait lekerekíthetjük, keretet készíthetünk, áttűnést – vízszintes, függőleges, sugaras és körkörös –, illetve számos egyéb effekttust csinálhatunk .

A tervező property editor-ában a következő stíluslap beírásokat add hozzá, vagy változtasd meg:

#### textinput és results list

background-color: white; border: 2px solid #8f8f91

Ez középszürke keretet készít.

#### groupbox

background-color: qlineargradient(x1: 0, y1: 0, x2: 0, y2: 1,

stop: 0 #bbbfc8, stop: 1 #e9eefd);

Vízszintes áttűnést készít.

#### Mindhárom label

background: transparent

Enélkül a label (címke) widget-ek keretét látnánk. Próbaképpen távolítsd el és nézd meg a különbséget.

#### A két button

QPushButton {
 border: 2px solid #8f8f91;
 border-radius: 8px;
 background-color: qlineargradient(x1: 0, y1: 0, x2: 0, y2: 1,
 stop: 0 #e9eefd, stop: 1 #bbbfc8);
QPushButton:pressed {
 background-color: qlineargradient(x1: 0, y1: 0, x2: 0, y2: 1,
 stop: 0 #bbbfc8, stop: 1 #e9eefd);};

A gomboknak keretet és lekerekített sarkot csináltunk. Még két áttűnést is alkalmaztam, egyet a gomb alapállapotához, egyet a lenyomott helyzethez. A button pressed (lenyomott) áttűnés a normál állapot áttűnésének fordítottja. Ez a 'top-to-bottom'-ot 'bottom-to-top'-ra változtatja, mozgás látszatát keltve.

A qt widget-ek testreszabásával kapcsolatban további információk találhatóak a http://doc.qt.io/archives/qt-4.8/stylesheetexamples.html címen.

Amikor kedvedre való a kinézet, akkor a stíluslapot add hozzá a sablonhoz, hogy megteremtsd az összes házi készítésű alkalmazásod egységes kinézetét és érzését. A következőkben látható a thesaurus mind világos, mind sötét témával. (következő oldal)

Ha az asztali környezetedben van konkrét kedven widget-stílusod, mint a Gtk+, a Plastique, vagy a Cleanlook, bármely widget-et, aminek a stílusát nem

### Alkalmi Python, 2. rész

írtad felül, ráveheted az adott stílus követésére, ha hozzáteszel egy ilyen sort:

QApplication.setStyle(QStyleFactory.create(' Cleanlooks'))

az \_\_init\_\_ method-ban a kódod elejénél, super(... ) és a self.setupUi(self) kifejezések közé.

#### És van még tovább is...

Most azt gondolhatnád, hogy egy grafikus szinonimaszótárat készítettünk. Nos igen, de ez egyben általános parancsvégrehajtó eszközt is python-ban és Qt5-ben. Majdnem bármelyik egyszerű parancssori eszközhöz könnyen átalakítható.

Például tegyük fel, egy ismert celeb nevében "szellemíróként" pénzért készítettél néhány írást, amit gyakran elő kell venned. Most már tudsz készíteni egy olyan egyszerű alkalmazást, ami kilistázza azokat egy szép, profin kinéző, grafikus ablakban.

A korábban általam említett fentről lefelé módszerben ez így néz ki:

Válaszd ki az egyik fő mesterművedet

Ehhez kell egy lista a dokumentumaidról

Listázd ki az összes elérhető dokumentumot

válaszd ki az egyik mesterművedet

Most ennyi iránymutatás elég, hogy honnan kezdjük el az alkalmazásunk módosításának kódolását.

Kell egy parancs, ami behozza nekünk a listát. Az összes anyagod a Documents könyvtárban van egy "wills\_plays" alkönyvtárban, LibreOffice Writer formátumban. Az is kell, hogy sorokba szedve



#### egyen

ként listázzon. A parancs, ami ezt teszi:

ls -1 /home/seth/Documents/wills\_plays # ez az egyes számjegy -1

A sablonunk használata nélkül újra hasznosíthatjuk a thesaurus alkalmazást, amivel még több időt spórolunk meg. Készíts egy "docz" könyvtárat. Másold ide a thesaurus.py-t, a thesaurus.ui-t és az update\_res.sh-t. A thesaurus-okat nevezd át docz.pyre, docz.ui-re és az update\_res.sh-ban a "thesaurus"okat írd át docz-re két helyen. Adj ikont a könyvtárhoz. Látod a rendszert abban, amit csinálunk? Ez a kidolgozási ciklus része. Szintén a kidolgozás során rendszeresen szerkesztjük designer-ben a felhasználói felületet, mentjük a változtatásokat és futtatjuk az update\_res.sh-t a változások alkalmazásához, szerkesztjük a pythonkódot és újraindítjuk az alkalmazást. A második részt újra végre kell hajtani, ahányszor csak változtattunk.

A tervezőben távolítsd el a fölösleges reset gombot, a line input mezőt és a "Result" címkét. Méretezd át a

result list mezőt a szabaddá tett részek felhasználására és cseréld le az ikont. A másik címkénél cseréld le az "Enter the search word"-öt "Select a documnet"-re. Mentsd a változásokat és futtasd az update\_res.sh-t.

A docz-t az alábbiak szerint változtasd meg. A megváltozott szöveget kiemeltem és töröltem néhány fölös sort, de általában a szükséges változtatások triviálisak és egyértelműek, mivel már egy kicsit többet tudunk a python nyelvről.

#!/usr/bin/env python3

import sys from PyQt5.QtCore import \* from PyQt5.QtWidgets import \* import subprocess import os import docz ui class Docz(QWidget, docz\_ui.Ui\_Form): def init (self): super(self.\_\_class\_\_, self).\_\_init\_\_() self.setupUi(self) self.quitButton.clicked.connect(self.exitApplication) os.chdir('/home/seth/Documents/wills\_plays/') cmd = "ls -1 \*.odt" # Thats ls -one not -ell result = subprocess.check output(cmd, shell=True).decode('utf-8') self.resultslist.setText(result)

def keyPressEvent(self, e):
 if e.key() == Qt.Key\_Escape:
 self.exitApplication()

def exitApplication(self): self.close() sys.exit()

if \_\_name\_\_ == '\_\_main\_\_': app = QApplication(sys.argv) form = Docz() form.show() app.exec\_()

•	Thesaurus	
	Select a document	
	All's Well That Ends Well.odt Antony and Cleopatra.odt As You Like it.odt Cornedy of Errors.odt Corribanus.odt Henry U, Part I.odt Henry IV, Part I.odt Henry V, Part II.odt Henry V, Part II.odt Macbeth.odt Merchant Of Venice.odt Merchant Of Venice.odt Much Ado about Nothing.odt Othelio.odt Pericles.odt Richard II.odt Romeo and Juliet.odt Taming of the Shrew.odt Tempest.odt Timus Andronicus.odt Troilus and Cressida.odt Twelfth Night.odt	
	Esc - Quit	

Hoppá! A tetején a sávban még Thesaurus olvasható. A javításhoz a designer property editorában a window title-t Docz-re írd át, ments és futtasd a update\_res.sh-t, majd indítsd újra az alkalmazást.

Ez az alkalmazás sem túl hasznos, de megmutatja a kód újrafelhasználásának előnyét. Most csak egy listát ad fel, de elérhetjük, hogy egér kattintást érzékelve megnyissa a dokumentumot Writer-ben és azt követően bezárja a docz alkalmazást. Ezt az ötletet fejlesztem tovább a következő részben, de előbb még egy kicsit többet meg kell tudnunk a python nyelvről. Ez következik majd.







## **Screenshot Showcase**



Posted by parnote, on February 16, 2019, running Xfce.

PCLinuxOS Magazine