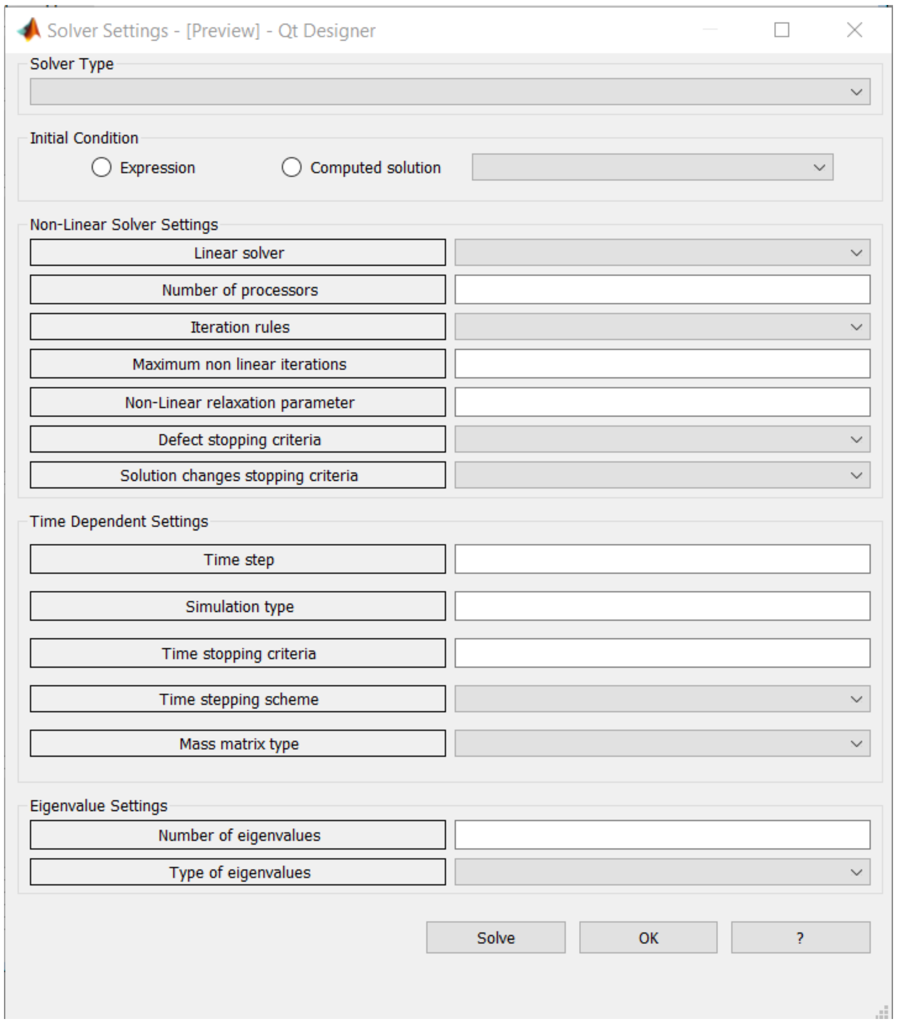
GUI using QT/Qml/Python

Participants	
Created By	
Type	
Created	@May 31, 2022 7:20 PM
Last Edited Time	@July 18, 2022 12:28 PM

Intro:

PyQt is a binding of Qt and python. The latest version is PyQt6. Qt is written in C++. PyQT enables us to use Qt framework(Written in C++) using Python. Qt is a large framework with many modules. We will focus on modules related with GUI.

QT Designer


Solver Settings - [Preview] - Qt Designer

Solver Type

Initial Condition

Expression

Computed solution

Non-Linear Solver Settings

Linear solver

Number of processors

Iteration rules

Maximum non linear iterations

Non-Linear relaxation parameter

Defect stopping criteria

Solution changes stopping criteria

Time Dependent Settings

Time step

Simulation type

Time stopping criteria

Time stepping scheme

Mass matrix type

Eigenvalue Settings

Number of eigenvalues

Type of eigenvalues

GUI produced using pyqt5

PySide

PySide is a Python binding of the cross-platform GUI development toolkit Qt, currently developed by the Qt company under the **Qt for Python** Project. Pyside provides LGPL-licensed Python bindings for the **Qt 4**. It also includes complete toolchain for rapidly generating bindings for any Qt-based C++ class hierarchies. PySide Qt bindings allow both free open source and proprietary software development and ultimately aim to support Qt platforms. Pyside is a really useful framework used for developing cool looking Graphical User Interfaces easily for your python applications.

- Pyside 2 supported Qt5, pyside 6 supports Qt6

Is Pyside2(Qt for Python) and PyQt same?

- https://machinekoder.com/pyqt-vs-qt-for-python-pyside2-pyside/

When do we need Qml?

- QML is a user interface markup language. It is a declarative language for designing user interface–centric applications. Inline JavaScript code handles imperative aspects. It is associated with Qt Quick, the UI creation kit.
- QML is a user interface specification and programming language. It **allows developers and designers alike to create highly performant, fluidly animated and visually appealing applications**
- You will not need Qt creator if you work on QML

QTQuick

Qt Quick is a free software application framework developed and maintained by the Qt Project within the Qt framework. It provides a way of building custom, highly dynamic graphical user interfaces with fluid transitions and effects, which are becoming more common especially in mobile devices

What is Namespace? (Found it while making module directory for PyDracula project)

A namespace is a system that has a unique name for each and every object in Python. An object might be a variable or a method. Python itself maintains a namespace in the form of a Python dictionary. Let’s go through an example, a directory-file system structure in computers. Needless to say, that one can have multiple directories having a file with the same name inside every directory. But one can get directed to the file, one wishes, just by specifying the absolute path to the file.

Real-time example, the role of a namespace is like a surname. One might not find a single “Alice” in the class there might be multiple “Alice” but when you particularly ask for “Alice Lee” or “Alice Clark” (with a surname), there will be only one (time being don’t think of both first name and surname are same for multiple students).

On similar lines, the Python interpreter understands what exact method or variable one is trying to point to in the code, depending upon the namespace. So, the division of the word itself gives a little more information. Its

Name

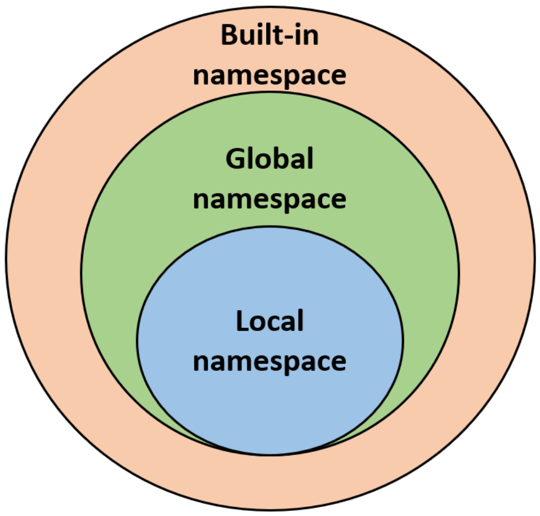
(which means name, a unique identifier) +

Space

(which talks something related to scope). Here, a name might be of any Python method or variable and space depends upon the location from where is trying to access a variable or a method.

Types of namespaces :

When Python interpreter runs solely without any user-defined modules, methods, classes, etc. Some functions like print(), id() are always present, these are built-in namespaces. When a user creates a module, a global namespace gets created, later the creation of local functions creates the local namespace. The **built-in namespace** encompasses the **global namespace** and the global namespace encompasses the **local namespace**.



Type of Namespaces

The lifetime of a namespace :

A lifetime of a namespace depends upon the scope of objects, if the scope of an object ends, the lifetime of that namespace comes to an end. Hence, it is not possible to access the inner namespace's objects from an outer namespace.

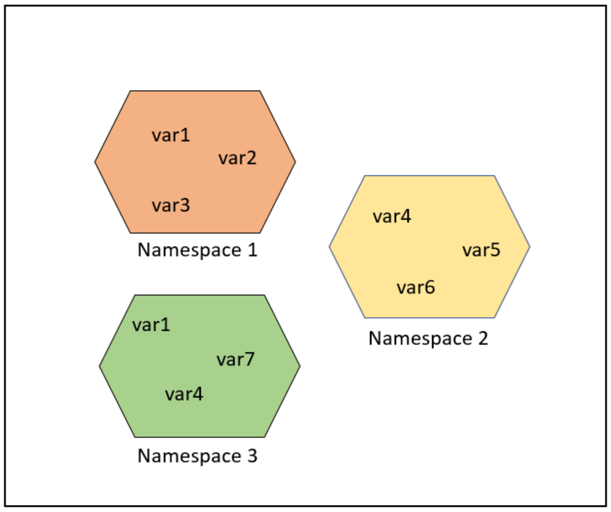
Example:

- Python3

```
# var1 is in the global namespace
var1 = 5
```

```
def some_func():
 # var2 is in the local namespace
 var2 = 6
 def some_inner_func():
 # var3 is in the nested local
 # namespace
 var3 = 7
```

As shown in the following figure, the same object name can be present in multiple namespaces as isolation between the same name is maintained by their namespace.



But in some cases, one might be interested in updating or processing global variables only, as shown in the following example, one should mark it explicitly as global and the update or process.   Note that the line “count = count +1” references the global variable and therefore uses the global variable, but compare this to the same line written “count = 1”. Then the line “global count” is absolutely needed according to scope rules.

- Python3

```
# Python program processing
# global variable
count = 5
def some_method():
 global count
 count = count + 1
 print (count)
some_method()
```

Output:

```
6
```

QSS file?

Style sheet file used by Qt programs; contains definitions for the look and feel of **GUI** elements, including fonts, sizes, colors, layouts, and mouse-over behaviors; stores a library of styles that can be incorporated into a user interface.

QSS files are similar to **.CSS** files. They can be created with any text editor.

What is QRC file?

- qrc file is **an XML document that enumerates local files to be included as runtime resources**. It serves as input to rcc .
- To run a project we need not have resources file.
- But when we want to relook or modify our project in Qt, we need to have these qrc files which let Qt know about the resources.
- qrc files are made in Qt.

A good read!

https://doc.qt.io/qt-6/resources.html#:~:text=Qt Resource Collection File (.&text=qrc%20file%20is%20an%20XML,serves%20as%20input%20to%20rcc%20.

What is PyCache?

When you run a program in Python, the interpreter compiles it to bytecode first (this is an oversimplification) and stores it in the `__pycache__` folder. If you look in there you will find a bunch of files sharing the names of the `.py` files in your project's folder, only their extensions will be either `.pyc` or `.pyo`. These are bytecode-compiled and optimized bytecode-compiled versions of your program's files, respectively.

As a programmer, you can largely just ignore it... All it does is make your program start a little faster. When your scripts change, they will be recompiled, and if you delete the files or the whole folder and run your program again, they will reappear (unless you specifically suppress that behavior).

When you're sending your code to other people, the common practice is to delete that folder, but it doesn't really matter whether you do or don't. When you're using version control (`git`), this folder is typically listed in the ignore file (`.gitignore`) and thus not included.

Converting .ui file of Qt to .py file

.ui file is xml file and we can’t modify that to our need. We therefore convert this .ui file to .py file and modify according to our need.

To modify,

1. open cmd at the .ui file location. (UI file will be created inside the project folder)
2. A) Now inside cmd use: `pyuic6 -o output.py -x input.ui` **(Python file in pyqt6 format)**

   B) `pyside6-uic input.ui -o output` `.py` **(Python file in pyside6 fromat)**

   Here, pyuic6 is because we are using pyqt6. -o is followed by python file we want to get output as. And -x is followed by input file which is in .ui file.

   ### A good read!

   https://www.pythonguis.com/faq/pyqt6-vs-pyside6/

About Qt designer, creator, custom widgets, Plugins.

https://youtu.be/B0X5FOev9Lw

Qt stylesheet?

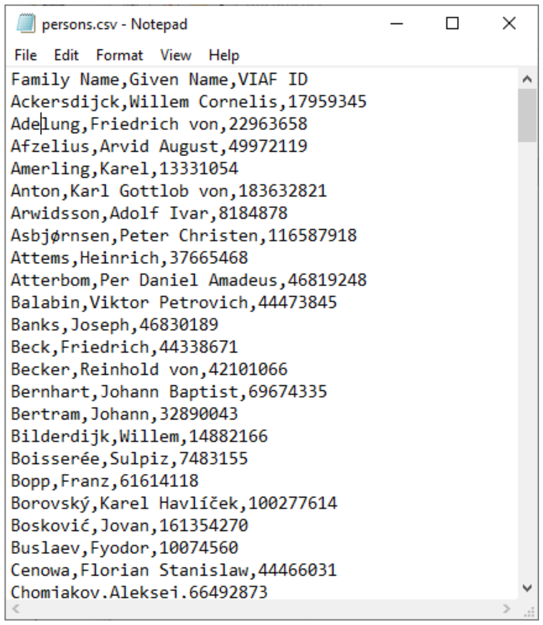
Qt Style Sheets are **a powerful mechanism that allows you to customize the appearance of widgets, in addition to what is already possible by subclassing QStyle.** The concepts, terminology, and syntax of Qt Style Sheets are heavily inspired by HTML Cascading Style Sheets (CSS) but adapted to the world of widgets.

- The widgets and labels are HTML aware and hence when we want to change the size, font etc we can directly use html script for changing it.
- In stylesheet we use CSS format.
- Using stylesheet we can create multiple types of GUI. We can enhance look and feel of our GUI using stylesheet. We can create themes like PyDracula using stylesheet.

CSV files:

A CSV file, as the name implies, typically separates information using commas. It's a way to exchange structured information, like the contents of a spreadsheet, among programs that can't necessarily talk to one another directly.

A typical CSV file looks like this:



```
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov.Aleksei.66492873
```

What is Venv?

venv (for Python 3) and virtualenv (for Python 2) **allow you to manage separate package installations for different projects**. They essentially allow you to create a “virtual” isolated Python installation and install packages into that virtual installation. Any module that we install, will be stored in venv/lib-packages/site-packages.

GitHub Actions:

GitHub Actions Tutorial - Basic Concepts and CI/CD Pipeline with Docker

Complete Github Actions Tutorial | GitHub Actions CI/CD | GitHub Actions Docker Build and Push | GitHub Actions Demo ► Complete DevOps Bootcamp - full educat...

https://youtu.be/R8_veQiYBjI