

Gui (graphical user interface) with python

List of Best Python GUI Libraries

- 1- PyQt5
- 2- Python Tkinter
- 3- PySide 2
- 4- Kivy
- 5- wxPython

framework for mobile applications (IOS and Arduino) using python.

- 1- Kivy
- 2- BeeWare
- 3- Pyqtdeploy
- 4- Ren'Py

Sorting a list by multiple attributes

Methods to sort a list by multiple attributes with Python:

- 1- Using lambda function
- 2- Using itemgetter
- 3- Using attrgetter

[Sort a list of objects by multiple attributes in Python - GeeksforGeeks](#)

garbage collection

- languages with garbage collection:
(SML, OCaml, Eiffel, D, Go, Haskell, c#, RPL, java most scripting languages)
- languages without garbage collection:
(c++, c)
- Some languages, like Ada, Modula-3, and C++/CLI, allow both garbage collection and manual memory management.

To do multimap in python

we used defaultdict like that:

```
In [18]: from collections import defaultdict

In [19]: md=defaultdict(list)

In [20]: md["sara"].append("a")

In [21]: md["sara"].append("b")

In [22]: print(md)
defaultdict(<class 'list'>, {'sara': ['a', 'b']})

In [23]: type(md)
Out[23]: collections.defaultdict
```

Assembly language:

- Extremely tiny microcontrollers which lack enough code space to meaningfully support a higher level language. If your whole program has to fit in a few hundred instructions, you may as well write those few hundred instructions.
- Time-critical code that can't be coded effectively in a higher level language. This could be a highly optimized "money loop;" that is, a key performance-critical loop for your application. Or, it could be code that needs to have very specific timing. It can't run too slow or too fast.
- Low-level machine manipulation that isn't supported by higher level languages. This can include accessing special machine specific registers, using specialized instructions, and so on. For example, most processors have a particular way to trigger a system call. Both the system call and handler entry point are typically written in assembly.
- to directly manipulate hardware, access specialized processor instructions, or evaluate critical performance issues.

Advanced for c++:

In this link: [Advantages and Disadvantages of C++ - GeeksforGeeks](#)

Difference between design and architecture patterns

- Architecture comes in Designing phase and Design Patterns comes in Building phase.
- Architectural pattern is like a blue print and design pattern is actual implementation.
- Architecture is base which everything else adhere to and design pattern is a way to structure classes to solve common problems.
- All Architecture is design pattern but all design pattern can not be architecture. Like MVC can come under both. But singleton design pattern can not be an architecture pattern. MVC, MVVM all come under both.
- Architecture : how components should behave and communicate in the system, set the physical location of components and finally choose the tools in order to create components. Design : while architecture deals more with the wide picture, design should drill down into the details relating to implementing certain components. Designing of components end up with classes, interfaces, abstract classes and other OO feature in order to fulfil the given component tasks.

My source:

- [Differences between Architecture and design pattern | by Divesh Singh | Medium](#)
- [الفرق بين ال design pattern و Architecture pattern - YouTube](#)

Applications of Hash Maps

- Problems related to item frequency.
- Dictionary
- File systems
- Password verification
- Storage optimization in the cloud
- Compiler operation

My source: [The Importance of Hash Maps. Hash maps can solve a whole bunch of... | by Supriya Sirbi | Better Programming](#)