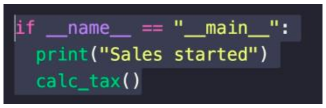
- In real world scenarios, we don’t put all code in one file instead we have folders structures hierarchy and several files
- SuperMarket ➔ Example of Sections → Edibles, Drinks etc
- A module is a file which contains some python code
- Naming conventions for files or modules is same as naming variables
- From sales here we just have the name and not the extension like sales.py
- If we press Ctrl + Space we will be able to see which objects the other module contains
- Don’t blindly import all objects from a module using asterisk *, import only those which you need
- We can import a whole module or some objects from that module, there is no best practice and no performance issues as if we need only a single object, still the whole module needs to be loaded
- __Pycache__ includes the (cpython file) compiled versions of all the module we import into this program
- Python stores this compiled version to speed up module loading speed (Note: not the performance of application)
- Python checks the compatibility of compiled file as it’s up to date or not by date etc
- Cpython file contains the python byte code
- Python doesnot create cache of the main file which is the entry point to that progam which we load via command line
- When we import a certain module, python searches for it in all the program directories (the folders we have made) and all the default direcotries python has made during installation + if any new framework, package or library has been installed
- We can use sys→ path to print all the directories
- A package is a container for one or more python modules
- We create __init__.py to declare a folder as a package
- A package is mapped to a folder and a module is mapped to a file
- Sub-Package is a sub-folder
- In intra package references we use absolute and relative ways of references
- Best Practice is absolute way
- The Built-in dir function, with dir we can get the list of methods and attirbutes defined in an object
- The name of module which starts our program is always __main__
- With this code you can use your modules as scripts



```
if __name__ == "__main__":
  print("Sales started")
  calc_tax()
```