**Why python?**

**How it is different from other programming languages?**

* Can build all kind of apps, agnostic of platforms and devices where the apps run.
* To deal with complex data- it has data structures to handle structured and unstructured data
* Makes programming easier, with lot of BIFs

dir(\_\_builtins\_\_)

to know abt particular bif, help(fn)

ex bif, isinstance(data, type) – isinstance(item, list)

* Reuability : Instead of repeating the code create functions for reusability

def fn\_name(args):

Suite is a block of code , say fn block , etc.

* Sharable code

Modules are easier to share

Code and build as a module and share to the python world.

The Python Package Index (or *PyPI* for short) provides a centralized repository for third-party python modules on the Internet. When you are ready, you’ll use PyPI to publish your

module and make your code available for use by others.

Import module name, to know the path of the module where the python interpreter’s looks for, modulename.path

* Exception handling

Try, except – pass is an null/empty statement – does nothing (reports runtime error and executes)

Finally – executes a suite in all conditions, when exception occurs r not.

* Classes – class classname: , init(), self- frst arg of a method inside a class (helps to identify instance/object)

Object creation – var = classname()

* Interpreter vs compiler

Complier and interpreter both translates source code to machine code, where compiler stores object code/executable as an intermediate code and interpreter doesn’t, it executes directly. Interpreter assigns values/allocates memory at runtime.

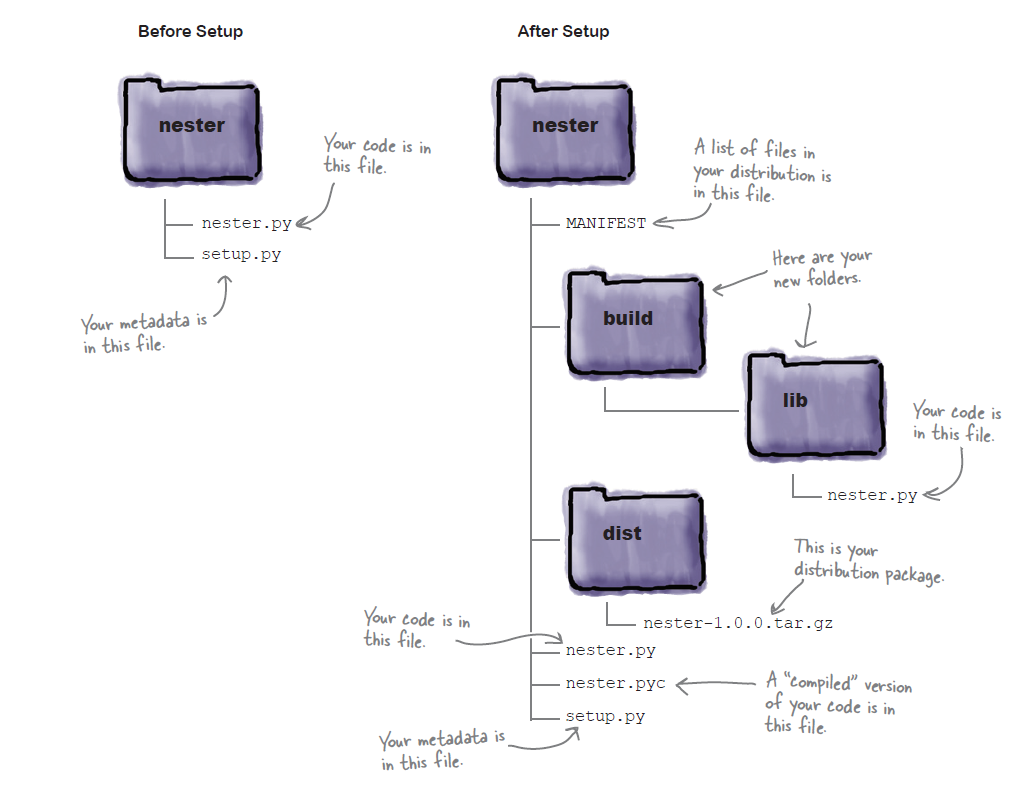
Compiler requires more memory than interpreter.

**Share modules**

Prepare distribution - put the .py file (module fn code) into the dir and also a create a setup.py file tat contains metadata of a module

Build distribution - python3 setup.py sdist

Install - python3 setup.py install



Yes, namespace names are like family names. When you want to refer to some function from a module

namespace other than the one you are currently in, you need to qualify the invocation of the function with the module’s namespace name.

So, instead of invoking the function as print\_lol(cast) you need to qualify the name as nester.print\_lol(cast). That way, the Python interpreter knows where to look. The format for *namespace qualification* is: the module’s name, followed by a period, and then the function name.

To import specific fns from a module, use from modulename import func. Ex: from nester import print\_lol.

Upload distribution to pypi – register with pypi website, also do cmdline registration,

Python3 setup.py register

Upload to pypi – python3 setup.py sdist upload

**Python data structures**

* List

List are like arrays, ex: [“sri”,”mike”,”jack”]

Python has no data types, no type declaration required

U can directly assign data/ds to identifiers

Ex: names = [“sri”,”mike”,”jack”]

To access list data, use offsets/index along with it, ex: names[0] will print ‘sri’.

U can use built-in list methods/fns like len, append, remove, insert(pos, data)

List can hold mixed data types, numbers and strings, strings with quotes and num just like tat,

[“sri”,”school”,16]

* Dict- keys and values
* Tuple

Immutable list within {}, cannot be changed.

**Loops**

* For – interpreter worries abt state info of data
* While
* If

**Comments**

# - single line comment

“ ” ” – triple quote multi-line comments