1. Setting the context
2. As a trainer what are my goals

>> Teach the basics of Python in an informal funny way

>> Hope to get you motivated to start programming

>> Get you curious about the language

>> Ensure that you are attentive (no blank stares ☺)

1. Teaching style

>> Interactive / personalized style with lots of visuals

>> Please STOP me if you see lengthy paragraphs or find me talking for too long on a specific topic

>> Get all of you to write lots of programs

>> From Day 1, you will write code

>> Quizzes

>> Timeouts where I change the context completely to break the monotony

>> Not only will you write code (Assignments/Activities), you will have to modify someone else’s code

>> Since it is a big group, I will randomly check your code

>> Will try to use images, because our brain understands visuals more easily than text

>> DON’T USE I – WE WILL EXPLORE TOGETHER

1. What you need to do

>> Do the exercises

>> Make notes

>> Don’t just read what I say. Digest it slowly. Stop me if I am going too fast

1. About myself
2. What’s next for you

The Python Eco-system

>> Different ways of accomplishing the same task

>> Being open source, developers have created so many tools

>> Python is used for a variety of different purposes.

>> Rich standard library and third party modules

>> Python is designed to be extendible.

>> The core is small and extremely fast with a small footprint.

>> So if your needs are simple, there is no overhead involved at all

>> Python is meant to help you solve problems

>> Typically programs accept input and generate output

>> Serious programs have to manage data

>> when you install Python 3 on your computer, an interpreter is installed, too. This is the technology that runs your Python code.

>> With Python, you edit your code and save it, and run it immediately

>> Doesn’t the code get compiled at some point ?

>> : Yes, it does, but the interpreter does not expose this process to the Python programmer (you). All of the details are taken care of for you. All you see is your code running as IDLE does all the heavy lifting, interacting with the interpreter on your behalf.

>> Executing Code, One Statement at a Time

>> As a developer, you need to write code. How will you write this code ? Editors, of course !

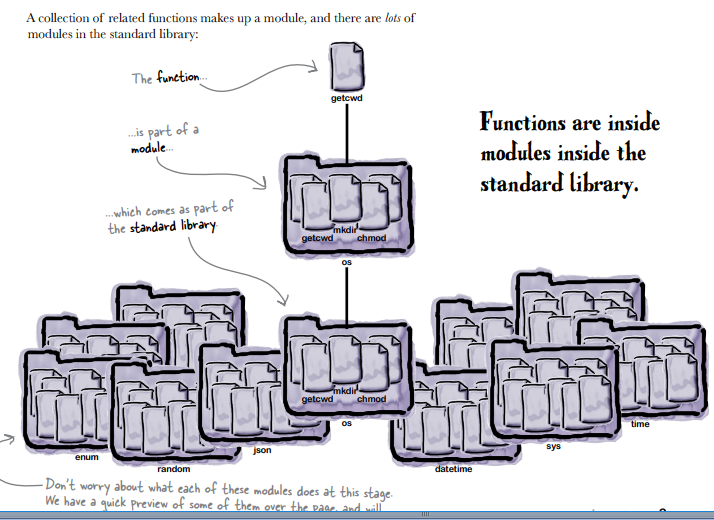
>> There are numerous editors that you can use. We will start with IDLE. This will keep you busy

>> Python’s interactive shell

* The first window, the Python Shell, is a REPL environment used to run snippets of Python code, typically a single statement at a time. The more you work with Python, the more you’ll come to love the Python Shell, and you’ll be using it a lot as you progress through this book. For now, though, we are more interested in the second window.

>> Do you want to introduce the Standard Library – YES

>> Include a diagram similar to this



>> Think of modules as a collection of related functions.

>> The standard library is the jewel in Python’s crown, supplying reusable modules that help you with everything from, for example, working with data, through manipulating ZIP archives, to sending emails, to working with HTML. The standard library even includes a web server, as well as the popular SQLite database technology. {{ This is copied text re-word }}

>> .The Python documentation has all the answers on the standard library. Here’s the kicking-off point:

https://docs. python.org/3/library/index.html

>> The standard library isn’t the only place you’ll find excellent importable modules to use with your code. The Python community also supports a thriving collection of third-party modules, some of which we’ll explore later in this book. If you want a preview, check out the community-run repository: <http://pypi.python.org>. {{ This is copied text re-word }}

>> THEME : Data Structures Come Built-in { I will refer to this repeatedly } The focus is on covering the course content, but given the power of data structures, we will explore more advanced concepts together

>> In our sample programs and assignments, we will at times be using the standard library

>> Please feel free to explore the standard library [[ Assignment for the group ]]

>> Python variables are dynamically assigned Before getting to the next line of code, perhaps a few words are needed about variables, especially if you are one of those programmers who might be used to pre declaring variables with type information before using them (as is the case in statically typed programming languages). In Python, variables pop into existence the first time you use them, and their type does not need to be predeclared. Python variables take their type information from the type of the object they’re assigned.

{{ How and When DO I introduce this concept }}

>> We will write programs and when required using the Interactive Shell

>> Let’s dive in

>> Let’s execute the simplest program possible using IDLE - Really-Simple.py

>> We could have done this directly from the Shell itself using the print function

>> The shell helps you run snippets of code

>> As you get more and more experienced, as a developer you will execute programs

>> One can also run this from the command line

python .\sample-programs\Really-Simple.py

>> More on the Interactive Shell

>>> help() {{ The prompt at the interactive shell changes to >>>, remember this }}