Book: Python crash course: a hands-on, project-based introduction to programming / by Eric Matthes.

The language 'Python' is named after Monty Python, not the snake. Guido van Rossum, the creator of Python, chose the name because he was a fan of "Monty Python's Flying Circus," a popular comedy show, and he wanted a name that was short, unique, and slightly mysterious.

Why Python?

o Python is an incredibly efficient language: your programs will do more in fewer lines of code than many other languages would require.
o Python's syntax will also help you write "clean" code.
o Python is used for many purposes: to make games, build web applications, solve business problems, and develop internal tools at all kinds of interesting companies. Python is also used heavily in scientific fields for academic research and applied work.

A **method** is an action that Python can perform on a piece of data. The dot (.) after name in name.title() tells Python to make the title() method act on the variable name. Every method is followed by a set of parentheses, because methods often need additional information to do their work.

➤ **Python** is a cross-platform programming language, which means it runs on all the major operating systems.
➤ In Python, the hash mark (#) indicates a comment.
➤ Python comes with an interpreter that runs in a terminal window, allowing you to try bits of Python without having to save and run an entire program.
➤ A **list** is a collection of items in a particular order.

- Python refers to values that cannot change as immutable, and an immutable list is called a **tuple.**
- A **tuple** looks just like a list except you use parentheses instead of square brackets.
- The **input()** function pauses your program and waits for the user to enter some text.
- A function that's part of a class is a **method.**
- The **__init__()** method is a special method Python runs automatically whenever we create a new instance of class.
- A **module** is a file ending in .py that contains the code you want to import into your Functions.
- The **super()** function at x is a special function that helps Python make connections between the parent and child class.
- **Exceptions,** a special Python class designed to help you respond to errors when they arise.
- Whenever an error occurs that makes Python unsure what to do next, it creates an exception object.
- **Refactoring** makes your code cleaner, easier to understand, and easier to extend.
- **Testing** proves that your code works as it's supposed to in response to all the input types it's designed to receive.
- A **test case** is a collection of **unit tests** that together prove that a function behaves as it's supposed to, within the full range of situations you expect it to handle.
- A **unit test** verifies that one specific aspect of a function's behavior is correct.
- The **Python Standard Library** is a collection of modules and packages that come bundled with Python, providing a wide range of functionality out-of-the-box.
- **pip** is a program that handles the downloading and installing of Python packages for you.
- **matplotlib,** a mathematical plotting library.
- The Zen of Python:

>>> *import this*

The Zen of Python, by Tim Peters

Projects:

1. Alien Invasion: Making a Game with Pytho
2. Data Visualization
3. Web Application

General Points:

❖ It's a good habit to keep an "ideas" notebook that you can refer to whenever you want to start a new project.
❖ It's a good idea to make the name of your list plural, such as letters, digits, or names.
❖ The **__pycache__** directory is automatically created by Python to store compiled bytecode files (.pyc and .pyo) that are generated when Python scripts are imported and run.
❖ A **histogram** is a bar chart showing how often certain results occur.
❖ When beginning a project, you first need to describe the project in a specification, or spec. Then you'll set up a virtual environment to build the project in. Like any good project or business plan, a spec should keep you focused and help keep your project on track.
❖ A virtual environment is a place on your system where you can install packages and isolate them from all other Python packages.