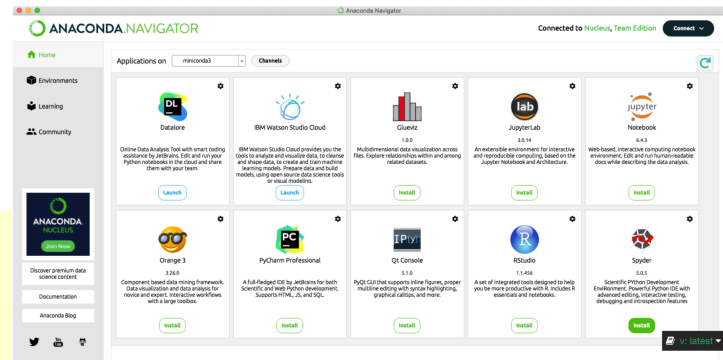


# Glossary of Programming Jargon

## Lesson 1

### Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) that allows you to launch applications, like Jupyter Notebook, without using the *command line* (see below).



### Bash

Bash is the shell (see below), or command language interpreter. It is a programming language through which a user communicates with the operating system or an application. When you type commands into the Terminal, you are writing bash script.

### Cells

A code cell in Jupyter notebook allows you to edit and write new code. When a code cell is executed, code that it contains will be sent to the *kernel*, the “computational engine”, which by default runs Python code. The results that are returned from this computation are displayed in the notebook as the cell’s output. The output can range from text to visualization (plots, tables, figures etc.).

### Code

Code is the set of instructions or a system of rules written in a particular programming language.

### Command Line

The command line is a user interface that is navigated by typing commands into a prompt (known as a *Terminal* or Command Prompt), instead of using a mouse. Unlike a graphical user interface (GUI), like your Desktop for example, a command line uses only a keyboard to navigate and not a mouse. When navigating your computer with a mouse, the computer is making a lot of assumptions in the background. By using the command line to tell your computer what you want to do, you are able to control every assumption the computer makes and action it takes.

### Compiled

The process of compiling a program occurs when you’re writing code in a compiled language. Compiling is required when you’re using a language like C or Java and not when you’re using an *interpreted language* (see below) like Python. The process of compiling is when you take the human readable code, the one which you or another programmer has typed up, and use a compiler, usually whatever the default compiler program is on your system, to turn it into machine readable code. Once you start compiling your code,

your compiler will tell you what errors exist either through a set of warnings or through an output file. It will not compile until all the errors are fixed. This machine readable code will then be available to you as an *executable* (see below).

## Dynamic Check and Static Check

The opposite of a *compiled* language is an *interpreted* one (see below). In this type of language, errors are caught as they are typed in a process called dynamic checking as opposed to a compiled language in which all the errors are reported at the time of compilation in a static check.

## Executable

An executable is what the machine readable version of a compiled code looks like. This is the file you run when you're ready to execute your code. If you want to modify your code after you've already created the executable, you edit the source code, the human readable version, and recompile it to create a new executable.

## Function

A function is a block of code that only runs when it is called. Functions can take input arguments and return some kind of output value. You can call a function anywhere within your code after you have defined it. Functions make code repeatable and testable.

## Interpreted

An interpreted language, like Python, is the opposite of a compiled one. These languages have interpreters that work in the background so that a program that you write can be directly executed in that language without requiring it to have been previously compiled into a machine readable language. Fun fact – the interpreter for Python is written in C!

## Jupyter Notebook

Jupyter Notebook is an open source (freely available to everyone) web application that you can use to create and share documents that contain live code, equations, visualizations, and text. You can open a Jupyter Notebook from Terminal or an application like Anaconda Navigator. Notebooks are useful for testing small amounts of code and analyzing data both visually and numerically. Most code that require a lot of computing power will likely be run directly on a computer since Notebooks cannot store much in their memory.

## LaTeX

LaTeX is a software system for document preparation. When writing, the writer uses plain text as opposed to the formatted text found in word processors like Microsoft Word. Similar to a compiled language, you write code in TeX that defines how you want your document to look, where you want text and figures to go, what the text should say, etc. and then you compile your code. This will render your code as a formatted text document (like a PDF) that follows the instructions you specified.

## Machine Learning

Machine learning is a branch of artificial intelligence and computer science that uses data and algorithms to imitate the way a human brain learns. As you train your algorithms by telling them when they're performing as expected versus when they are inaccurate, they follow a learning curve and gradually increase in accuracy. Machine learning techniques, when adequately accurate, can replace the decision making of a human being or make efficient predictions for data trends.

## Markdown

Markdown is a simple set of rules to format and style plain text. You can write in markdown in Jupyter Notebook cells and the cell will convert it into HTML text. Alternatively, you can also directly write in HTML in a markdown cell in Jupyter Notebook.

## Modules

In Python, modules are all files with the ".py" extension containing Python code that can be imported inside another program. A module can refer to a code library or a file that contains a set of functions that you want to include in your programs..

## Operating System

An operating system is the most important software that runs on a computer. It manages the computer's memory, processes, software and hardware. It also allows you to communicate with the computer without knowing how to speak the computer's language. When you directly want to speak to the computer, you use the command line, in all other cases, you go through your operating system. Three of the most popular operating systems are Microsoft Windows, macOS (for Apple), and Linux.

## Program

A computer program is a sequence of instructions written in any programming language that are meant for a computer to compile or interpret.

## Session

A session is an instance of programming that only exists temporarily. For example, any variables you define in a terminal session from the command line will disappear after you close the terminal window, unless you write those variables to a saved file. .

## Shell

The shell is a program that takes commands from the keyboard and gives them to the operating system to perform. It is how one communicates with the operating system. On most Linux systems, like Mac, a program called Bash acts as the shell program.

## Software

Software is the collection of programs and routines that make a computer or electronic device run. Any code you write is considered software.

## Source Code

Source Code is the human readable version of code. For compiled languages, it is the code that you send to the compiler to turn into an executable. For interpreted languages, all code is source code.

## Syntax

Syntax is a set of rules for grammar and spelling. In other words, it means using character structures that a computer can interpret. For example, if a user tries to execute a command without proper syntax, it generates a syntax error which causes the program to fail. Common syntax errors include misspelled words, missing keywords, missing commas, missing parentheses, and misused mathematical operators. Ex. Misspelling the keyword `for`:

```
In [3]: 1 fro i in range(10):
        2     print(i)

File "<ipython-input-3-4429cd38d3ca>", line 1
    fro i in range(10):
      ^
SyntaxError: invalid syntax
```

## Terminal

This is a program that opens a window and lets you interact with the shell. Here, is where you will talk to the computer using the shell to either write code, navigate to existing code, run code, and otherwise interact with the contents of your computer.

## Text Editor

A text editor is a computer program that edits plain text. A common text editor that is provided with most operating systems is "Notepad." Whereas the Notepad text editor has a graphical user interface (GUI) that allows you to use your mouse and visualize your text graphically, you can also edit text directly from the command line using text editors like Emacs or Nano. You can write and edit code written in any programming language using a text editor.

## Type-Checking

Type checking is testing a program for "type errors" either during the time of compilation (static check) or during program execution (dynamic check). It ensures that operands are of a compatible type. A compatible type is one that is allowed for the operator or follows the programming language rules that can forcefully convert it to be compatible. For example, in `c = a + 3*b`, `b` must be a type that can be multiplied by an integer. If `b` were a text type, a string, this operation would fail as `b` would be an incompatible type.

## Visualize/Visualization

Data visualization is the graphical representation of information and data. Data visualization tools include visual elements like charts, graphs, maps, tables, etc. in order to provide an accessible way to see and understand trends, outliers, and patterns in data.

## Web-Development

Web-Development is the process of building, creating, and maintaining a website.

## Whitespace

Whitespace is any character or series of characters which creates space on a page but does not leave any visible marks. Commonly, whitespace is created with the use of tabs and spaces.

It is closely related to indentation which is a strategic use of whitespace to enhance the readability of code. For example, when you write a loop in Python, the first line of your loop starts from the left most alignment, the loop begins with the second line which is tab-indented to be nested underneath the first line. Anything that follows the same indent pattern will be considered part of that loop.

Notice how changing the indent changes the output of the following for loop:

In [9]:

```
1 for i in range(5):
2     print(f'i = {i}')
3
4     for j in range(2):
5         print(f'j = {j}')
```

```
i = 0
j = 0
j = 1
i = 1
j = 0
j = 1
i = 2
j = 0
j = 1
i = 3
j = 0
j = 1
i = 4
j = 0
j = 1
```

In [10]:

```
1 for i in range(5):
2     print(f'i = {i}')
3
4 for j in range(2):
5     print(f'j = {j}')
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
j = 0
j = 1
```

## *Lesson 2*

### Immutable

Many Python objects are immutable. This means that after you create the object and assign some value to it, you cannot modify that value.

### Libraries

Python libraries are a set of useful functions such that you do not need to write codes from scratch. Some of the most commonly used libraries for Astronomy include Numpy (mathematical library), Scipy (algorithms for optimization, integration, statistics and more), Astropy (tools specifically for Astronomy), and Matplotlib (tools for data visualization).

### Types

Variables can store data of different types, and types can do different things. In order to perform operations between different variables, types must be compatible. Types include text, numeric, sequence, mapping, boolean, set, and binary.

### Iteration

Programming algorithms consist of steps that are performed one after another. Sometimes these algorithms need to repeat certain steps until told to stop or until a particular condition has been met.

Iteration is the process of repeating steps.

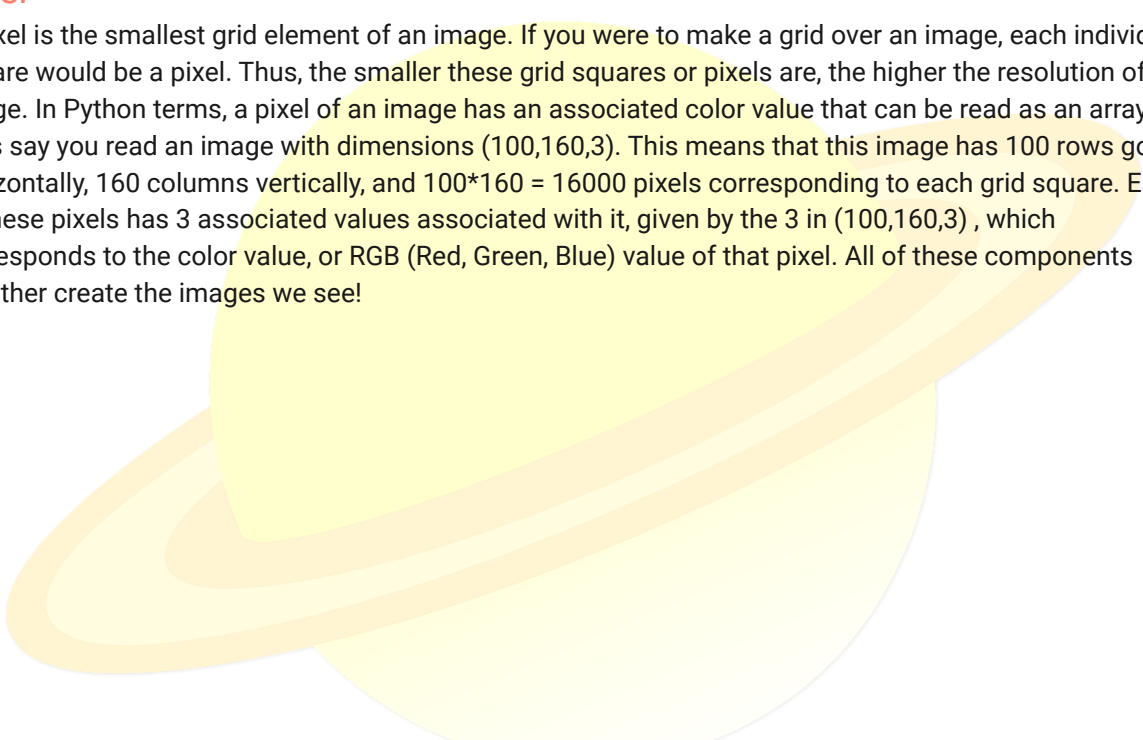
## Lesson 3

### Array

Python arrays are a data structure like lists. The values in an array can be accessed by using a row – column format, with up to 32 dimensions. They are a tool to store multiple items of the same type together. Python arrays can be iterated over and have a number of built-in functions to handle them. A way to visualize arrays is by thinking of a 2D instance of an array like an image. An image is essentially a 2D array with a color value associated with every row – column unit.

### Pixel

A pixel is the smallest grid element of an image. If you were to make a grid over an image, each individual square would be a pixel. Thus, the smaller these grid squares or pixels are, the higher the resolution of the image. In Python terms, a pixel of an image has an associated color value that can be read as an array. Let's say you read an image with dimensions (100,160,3). This means that this image has 100 rows going horizontally, 160 columns vertically, and  $100 \times 160 = 16000$  pixels corresponding to each grid square. Each of these pixels has 3 associated values associated with it, given by the 3 in (100,160,3), which corresponds to the color value, or RGB (Red, Green, Blue) value of that pixel. All of these components together create the images we see!



## *Lesson 4*

### Data Visualization





## *Further References*

<https://www.w3schools.com/python/default.asp>

