General Assembly

STUDY GUIDE

DATA TYPES IN PYTHON

Key Terms and Definitions

» **Python:** One of the most popular programming languages for data science.

» **REPL (Read, Evaluate, Print, Loop):** Allows the user to type and evaluate one line at a time. It reads the user's input, evaluates it, prints the output, then loops back to prompting the user for input.

» **Python Enhancement Protocols (PEPs):** Documents that explain the "why" behind each new Python feature. A new PEP must be written for the community to consider a new feature.

» **Expression:** Any valid piece of code that evaluates to something.

» **Operators:** Pieces of Python code that act on one (unary operators) or two (binary operators) objects and evaluate to a single object. Some examples include:

- =: The assignment operator, which associates a name with a value to create a variable. It is not a comparison, it's a definition.
- []: The indexing operator used to get a particular part of the variable (a character in a string, item in a list, etc.).
- ==: The comparison operator, which tests whether two objects are identical and evaluates to either True or False.
- %: The modulus operator, which returns the remainder of the division of two objects.

» **Data Types:** Indicators of how a computer should evaluate a particular object. For example, the same binary sequence (set of 1s and 0s) can evaluate to either a number or letter, depending on its type. The complete list of Python's built-in data types is: none, Booleans, numbers (int/float/long/complex), strings, lists, tuples, sets, and dictionaries.

» **Primitive Data Types:** Data types that are single objects and are immutable (cannot be changed without being redefined). These are:

- *Integers:* Whole numbers (with an optional + or - prefix).
- *Floats:* Decimal numbers (formally called floating point numbers).
- *Booleans:* Also called flags; these can only be True or False.
- *Strings:* Sequences of characters, always enclosed in quotation marks.

» **Constructor:** A built-in function of Python data types that assists with creation or conversion of these objects.

» **Objects:** Everything in Python is an object, which is just a particular grouping of functions and variables.

- **Methods:** Functions that "belong to" an object and act on its variables.
- **Attributes:** The variables within an object.
- **Indexing:** Referencing a character or item within an object. Python uses zero-based indexing, which means the first item is referenced by index 0 and the last by one less than the number of characters.

Guiding Questions

1. Why must strings be enclosed in quotation marks?
2. Why are "primitive" data types named as such?
3. If you are thinking about a car as an "object," what would be some of its methods? Attributes?

Additional Resources

1. DataCamp: Intro to Python for Data Science
    - Section 1: "Python Basics." Specifically, see the parts on "variable types" onward.