Primitive types — int, float, str, booleans, etc.
Collection — Lists, Dictionaries, Tuples, Sets…

Records?

Collection of data fields

Data types with functions attached that only apply to that data type

s: str = “This is a string”
s.startsWith(…)

Functions that only apply to a specific data type are known as “Methods”

Functions -> return a value
Procedures -> do not a value
Methods -> only apply to specific data types

Object-Oriented Programming

Procedural Programming -> . up until the late 1980s early 1990s

Data had no protection.

C -> C++ (C with classes)

Objective-C -> Swift

SmallTalk circa 1967

Encapsulation -> brick wall + methods

Bank Account. Directly change the balance and the history of transactions = NO

withdrawn, deposit, …

Internals of the “object” are hidden from the outside world.
Data protection is therefore provided.

“Cookie cutter” -> make an int. Make a BankAccount?
Variables of a specific type (e.g., int)
“Instances” of a “Class”, these “instances” are called “objects”.
Template of the desired structure.

Encapsulation

Inheritance ->. Vehicle -> Car isa Vehicle
Add additional functionality to the Vehicle class to get the Car class.
Don’t copy the items down the “system” does this for you.

Polymorphism

len(str)
len(list)
len(dict)

len works with multiple types and selects the appropriate method to execute
"Dispatching on type”
if a str use the len for strings
if a list use the len for lists
…

Message Passing

obj1 -> mess -> obj2 -> obj3 -> obj2
asynchronous -> that don’t wait for a reply.
synchronous -> wait for a reply

Google Search

Sends off many agents to look for answers.
It segments up the “database” so that different processes can search concurrently.
This happens asynchronously.

3.13 -> Global Interpreter Lock (GIL)