Python Session

DataDNA

Class

- Class is a **user defined prototype or blueprint** for creating an object
- Everything in Python is an object, with its properties and methods
- Classes provide a means of packaging data and functionality together.
- Instantiating a Class create a new object through which we can access attributes of the Class
- The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

Class - Important Terminologies

Instance – An individual object of a certain class. An object obj that belongs to a class Circle, for example, is an instance of the class Circle.

Instantiation – The creation of an instance of a class.

Method – A special kind of function that is defined in a class definition.

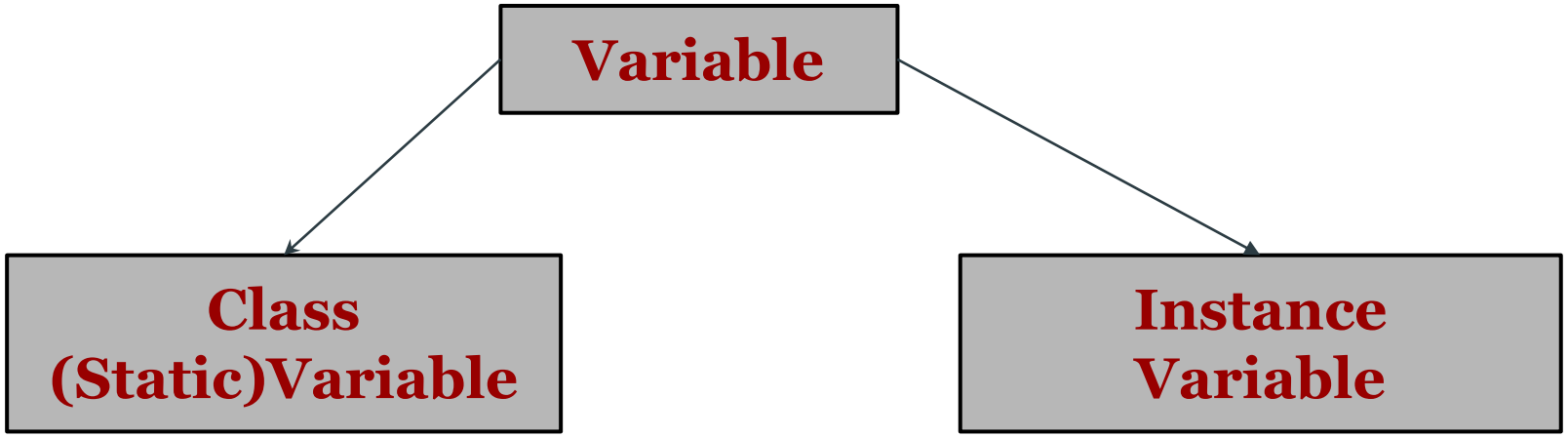
Method Creations: We can create other methods inside a class as normal functions with the fact that first argument in the method should be self.

Object – A unique instance of a data structure that's defined by its class. An object comprises both data members (class variables and instance variables) and methods.

__init__() - Special method, it is called Class Constructor or as Initialization method that Python calls when you create a new instance of class.

Data member – A class variable or instance variable that holds data associated with a class and its objects.

Class - Important Terminologies



- Class variable – Class variables are defined within a class but outside any of the class's methods.

  A variable that is shared by all instances of a class. The value of this variable will be same across all the objects of the class.

- Instance variable – A variable that is defined inside a method and belongs only to the current instance of a class.

  As the instance changes, these variables also change. Basically, this will be different for different objects of the class

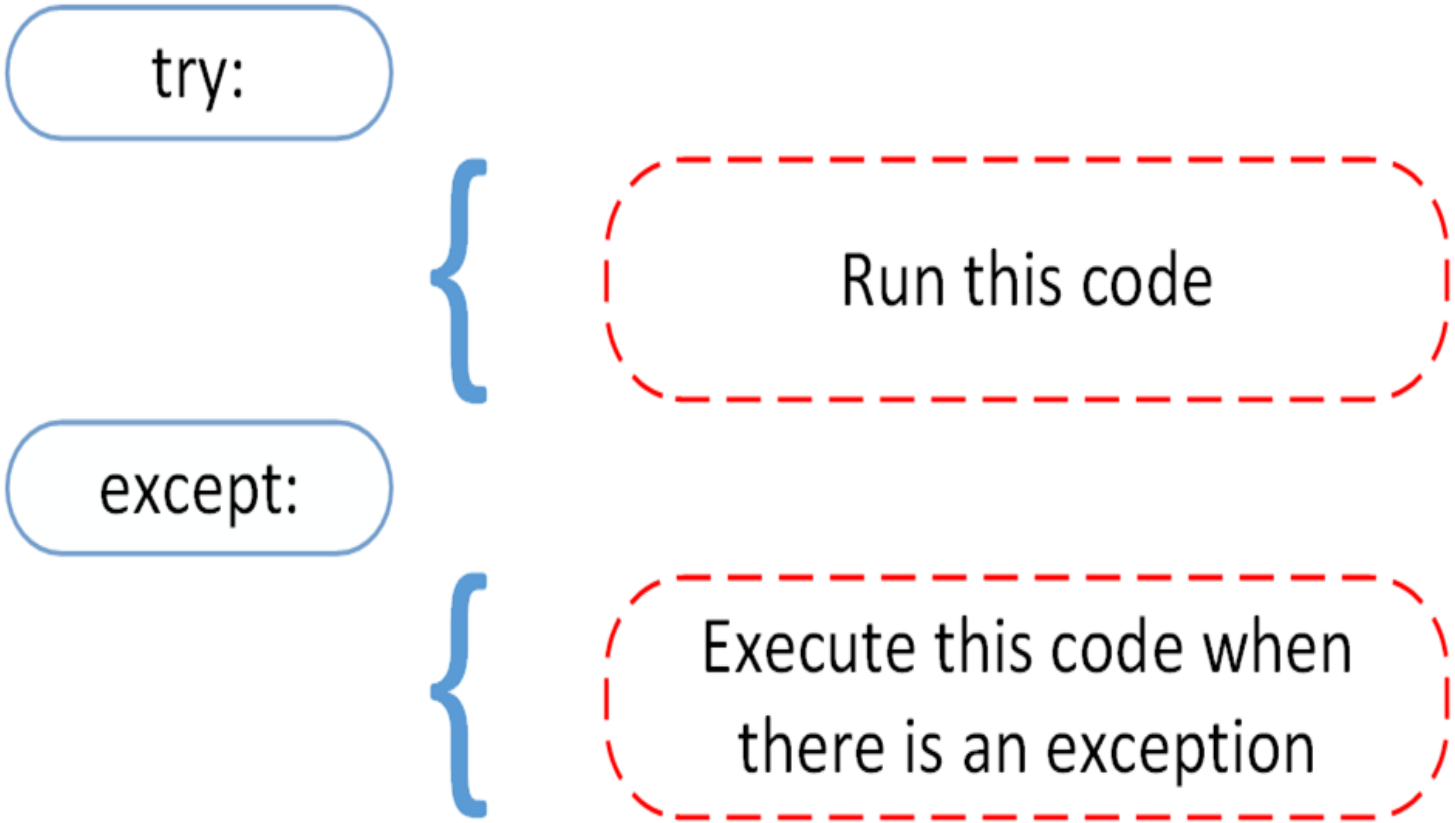
DataDNA

Class - Accessibility Checklist

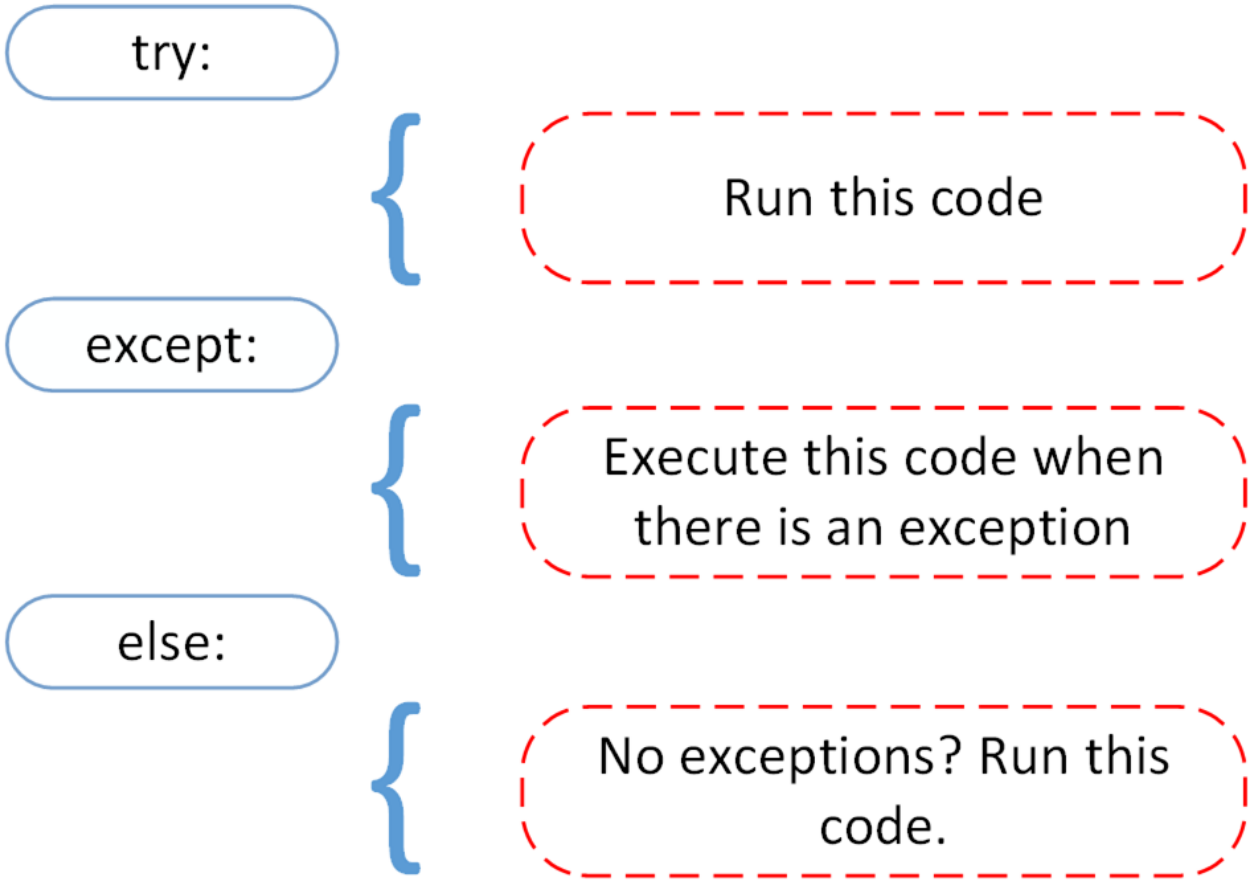
Variable Define and Initialize in	Accessible outside class	Accessible inside class	Accessible inside method	Accessible inside another class	Syntax	Example
Outside Class	Yes	Yes	Yes	Yes	Variable name	first_name
Inside Class	No	Yes	Yes	No	For Class - Variable namae For Method - Class Instance.variable nama	For Class - first_name For Method - self.first_name
Inside Method	No	No	Yes	No	Variable name	first_name

Exception Handling

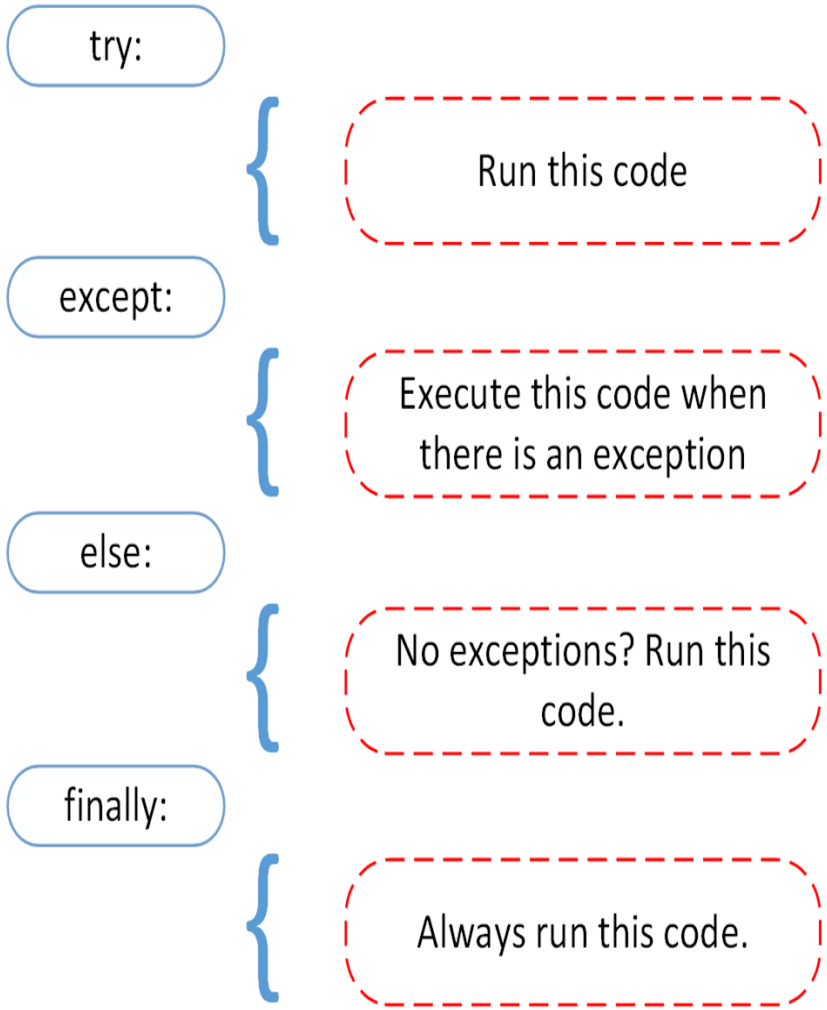
Try - Except


try:
Run this code
except:
Execute this code when there is an exception

Try - Except - Else



Try - Except - Else - Finally



Json Handling

Json Intro

Content Source: https://www.json.org/json-en.html

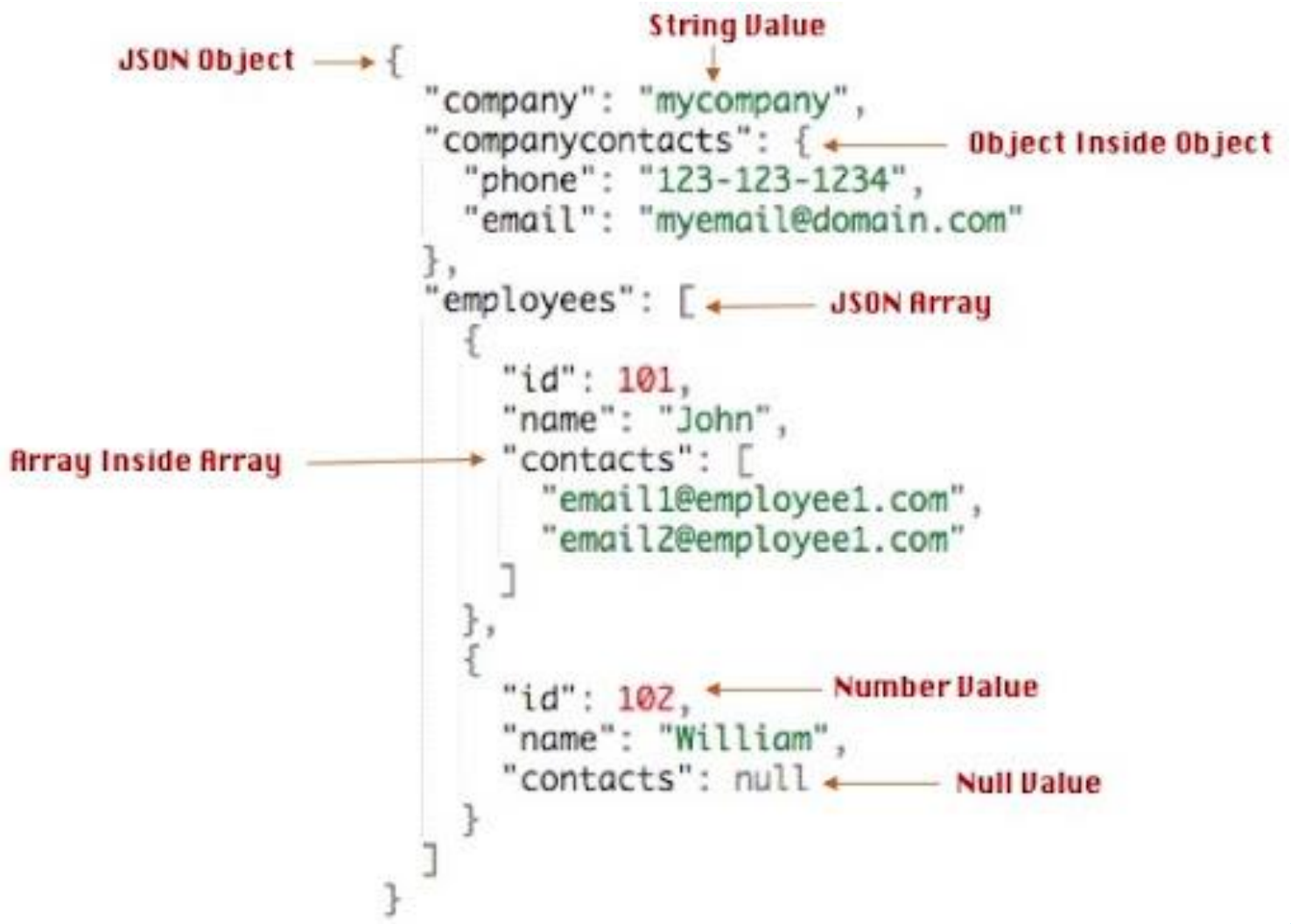
- JSON (JavaScript Object Notation) is a lightweight data-interchange format.
- It is easy for humans to read and write. It is easy for machines to parse and generate.
- It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999.
- JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

Json Structure

Content Source: Google images



Important Libraries

References

- https://towardsdatascience.com/exploratory-data-analysis-with-pandas-profiling-de3aae2ddff3
- https://towardsdatascience.com/intelligent-visual-data-discovery-with-lux-a-python-library-dc36a5742b2f