**Content**

[1. JSON and data types 2](#_Toc15923759)

[1. Intro into JSON 2](#_Toc15923760)

[2. Convert from JSON to Python 2](#_Toc15923761)

[3. Convert from Python to Json 3](#_Toc15923762)

[2. Iterators, yield, generators 4](#_Toc15923763)

[1. Generator functions 4](#_Toc15923764)

[2. Generator expressions 5](#_Toc15923765)

[3. Use Cases 5](#_Toc15923766)

1. JSON and data types
   1. Intro into JSON

In computing, **JavaScript Object Notation (JSON)** is an **open-standard file format** that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems.

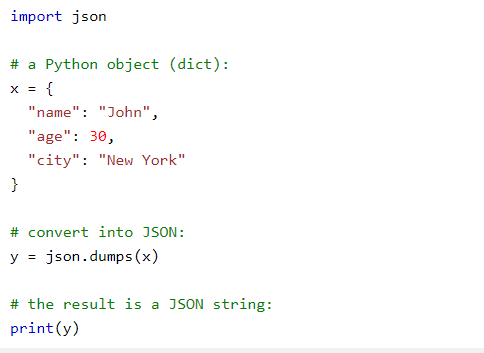
JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json.

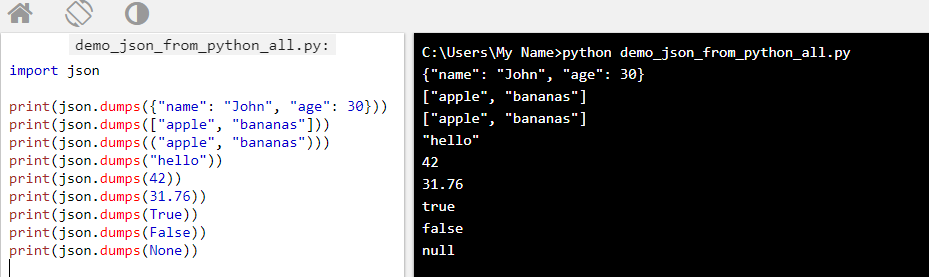
**Asynchronous JavaScript and JSON (or AJAJ)** refers to the same dynamic web page methodology as **Ajax**, but instead of **XML**, **JSON** is the data format. AJAJ is a web development technique that provides for the ability of a webpage to request new data after it has loaded into the web browser. Typically it renders new data from the server in response to user actions on that webpage. For example, what the user types into a search box, client-side code then sends to the server, which immediately responds with a drop-down list of matching database items.

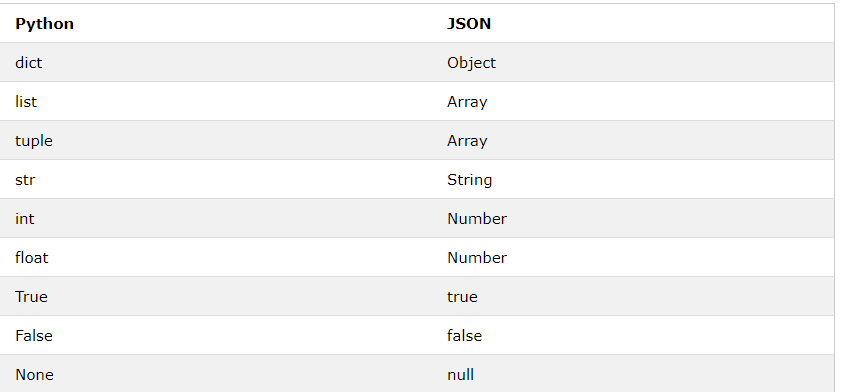
* 1. Convert from JSON to Python



* 1. Convert from Python to Json





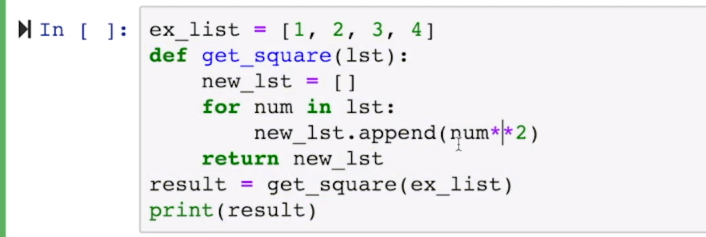


1. Iterators, yield, generators

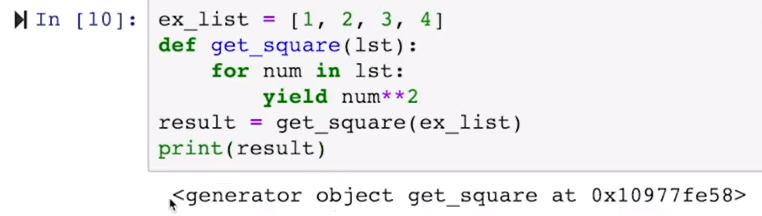
* Generator functions (yield)
* Generator expressions
* Optimizing the memory space.
* E.g. when working w large excel to not run out of memory (Lol 2 late  )
* Result set need not be constructed all at once
  1. Generator functions

Unlike normal functions, that receive an input and return a result immediately, generator functions produce result only when needed.

Python generators are able to suspend and resume their execution around the point of value generation



But we do not need the temporary list.



3 ways how to go through a generator object:

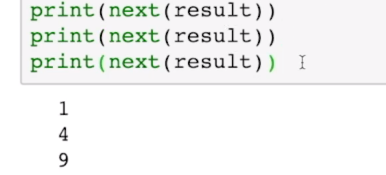
1. For loop



1. list built-in method to convert the iterator to a list



1. next method: every iterator object in python defines a next method. Returns the next item in the iteration till all of the items are over. Results in StopIteration error.



* 1. Generator expressions

List comprehension:

