### Contents

[w3schools 2](#_Toc13924960)

[Convert from JSON string to Python using json.loads() method 2](#_Toc13924961)

[Convert from Python to JSON string using json.dumps() method 2](#_Toc13924962)

[Convert Python objects into JSON strings, and print the values: 2](#_Toc13924963)

[Order the Result 3](#_Toc13924964)

[Format the Result 3](#_Toc13924965)

[JSON\_CoreyMS 4](#_Toc13924966)

[# loads() - Convert from JSON to Python 4](#_Toc13924967)

[# dumps() = Convert from Python to JSON 4](#_Toc13924968)

[# load() - loads JSON data from a file 5](#_Toc13924969)

[# dump() - method write JSON object to file 5](#_Toc13924970)

[Socratica 6](#_Toc13924971)

# w3schools

A JSON string is almost identical to a Python dictionary.

## Convert from JSON string to Python using json.loads() method

# some JSON:

x = '{ "name":"John", "age":30, "city":"New York"}'

# parse x:

y = json.loads(x)

# the result is a Python dictionary:

print(y)

## Convert from Python to JSON string using json.dumps() method

import json

# a Python object (dict):

x = {

"name": "John",

"age": 30,

"city": "New York"

}

# convert into JSON:

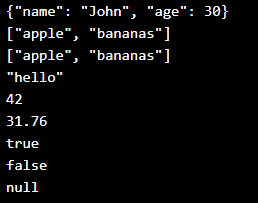
y = json.dumps(x)

# the result is a JSON string:

print(y)

## Convert Python objects into JSON strings, and print the values:

import json



print(json.dumps({"name": "John", "age": 30}))

print(json.dumps(["apple", "bananas"]))

print(json.dumps(("apple", "bananas")))

print(json.dumps("hello"))

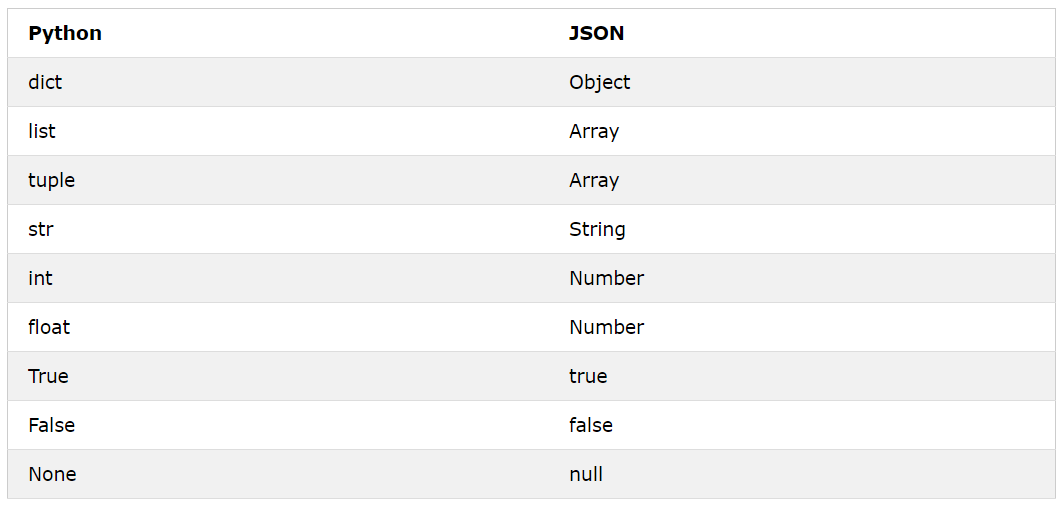
print(json.dumps(42))

print(json.dumps(31.76))

print(json.dumps(True))

print(json.dumps(False))

print(json.dumps(None))

When you convert from Python to JSON, Python objects are converted into the JSON (JavaScript) equivalent:

## Order the Result

Use the sort\_keys parameter to specify if the result should be sorted or not:

json.dumps(x, indent=4, sort\_keys=True)

## Format the Result

Use the **indent** parameter to define the number of indents.

Use the **separator** parameter to change the default separator.

You can also define the separators, default value is (", ", ": "), which means using a comma and a space to separate each object, and a colon and a space to separate keys from values.

json.dumps(x, indent=4, separators=(". ", " = "))

# JSON\_CoreyMS

import json

# multiline string

# a key of members and

# the value of members is an array

people\_string = """{

"members": [

{

"name": "Molecule Man",

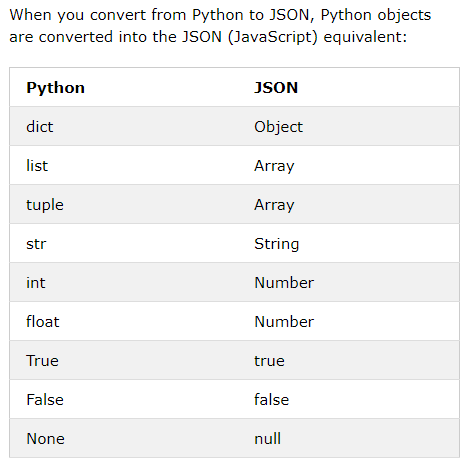
"age": 29,

"secretIdentity": "Dan Jukes",

"powers": [

"Radiation resistance",

"Turning tiny",

 "Radiation blast"

]

},

{

"name": "Eternal Flame",

"age": 1000000,

"secretIdentity": "Unknown",

"powers": [

"Immortality",

"Heat Immunity",

"Inferno",

"Teleportation",

"Interdimensional travel"

]

}

]

}

"""

## # loads() - Convert from JSON to Python

data = json.loads(people\_string)

print(type(data)) # <class 'dict'> # here JSON is a Python obj

print(type(data['members'])) # <class 'list'>

for item in data['members']:

del item['secretIdentity']

## # dumps() = Convert from Python to JSON

data\_str = json.dumps(data, indent=2, sort\_keys=2)

print(data\_str)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import json

## # load() - loads JSON data from a file

with open('states.json') as f:

data = json.load(f) # now data is a python object

for state in data['states']:

del state['abbreviation']

## # dump() - method write JSON object to file

with open('states\_area\_code.json', 'w') as f:

data2 = json.dump(data, f)

**Note:** The difference between dumps() vs dump, loads() vs load() is that dumps() and loads() operate on JSON strings, but dump() and load() operate on JSON files.

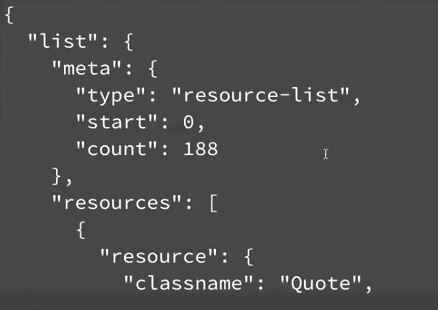
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import json

from urllib.request import urlopen

with urlopen("https://finance.yahoo.com/webservice/v1/symbols/allcurrencies/quote?format=json") as response:

source = response.read()



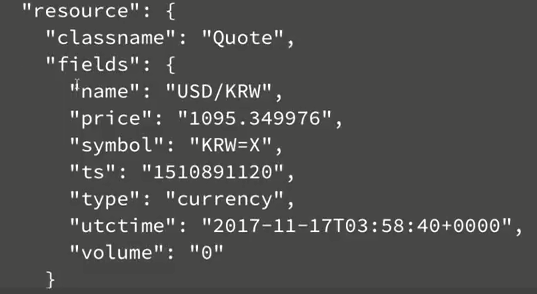
data = json.loads(source)

# print(json.dumps(data, indent=2))

usd\_rates = dict()

for item in data['list']['resources']:

name = item['resource']['fields']['name']

 price = item['resource']['fields']['price']

usd\_rates[name] = price

print(50 \* float(usd\_rates['USD/INR']))

# Socratica

|  |  |
| --- | --- |
| json.load(f) | Load JSON data from a file |
| json.loads(s) | Load & convert JSON string to Python |
| Json.dump(j, f) | Write JSON object to a file |
| Json.dumps(j) | Output and convert JSON string to Python |