

Python JSON Tutorial

Python JSON Tutorial

In recent times, JSON is preferred syntax for storing and exchanging data between applications.

In Python programming, you can use `json` package to work with JSON strings. In this tutorial, we will learn how to use json Python package and work with JSON data.

Import JSON

json package comes inbuilt with Python. So, all you have to do to use json package is, import at the start of your program.

To import json into your Python program, use the following import statement.

```
import json
```

Parse JSON String and Access Elements

Consider the following JSON String.

```
{"rollno":25, "name":"Raghu", "class":7, "section":"B"}
```

We will parse this JSON string using `loads()` function of json module. And we shall print the individual elements of the JSON object.

example.py – Python Program

```
import json

#json string
jsonStr = '{"rollno":25, "name":"Raghu", "class":7, "section":"B"}'

#parse/load json string into json object
jsonObj = json.loads(jsonStr)

#access
print(jsonObj["rollno"])
print(jsonObj["name"])
print(jsonObj["class"])
print(jsonObj["section"])
```

Output

25
Raghu
7
B

As you can observe, once the JSON string is parsed to an object `jsonObj`, you can access the individual elements with key used as index.

For example, `jsonObj["rollno"]` returns the value for the key `rollno` in `jsonObj`.

Python JSON – Access Inner Nodes

You can access inner nodes in the same way as that you access elements in multi-dimensional array.

In this example, we take a JSON string, in which one of the element `marks` has elements in it `{"science":87, "maths":34}`.

example.py – Python Program

```
import json

#json string
jsonStr = '{"rollno":25, "name":"Raghu", "marks":{"science":87, "maths":34}}'

#parse/load json string into json object
jsonObj = json.loads(jsonStr)

#access inner nodes
print(jsonObj["marks"]["science"])
```

Output

Parse JSON Array

You can parse a JSON array and access the elements using index. The index starts from `0` and increments for the subsequent elements as in a list.

In the following example, we have an array of two elements. We shall parse the JSON array using the same `loads()` function, and then access the array elements using index.

example.py – Python Program

```
import json

#json string
jsonStr = '[{"rollno":1, "name":"Prasanth"}, {"rollno":2, "name":"Raghu}]'

#parse/load json string into json object
jsonObj = json.loads(jsonStr)

#access the json array
print(jsonObj[0])
print(jsonObj[1])
```

Output

```
{'rollno': 1, 'name': 'Prasanth'}
{'rollno': 2, 'name': 'Raghu'}
```

Convert Python Objects to JSON

You can convert some of the types of Python Objects to JSON. Using this, you can save python objects directly to files in persistent storage. Or you can transmit them to other applications as JSON string, which is a good thing, because most of the applications are embracing JSON syntax for data transfer between the applications.

We can convert following Python objects to JSON String.

- Dictionary
- List
- Tuple
- String
- Integer
- Float
- Boolean
- None

In the following example, we shall convert some of the Python objects to JSON String.

example.py – Python Program

```

import json

#list to json string
alist = [12, 57, 41, 68, 47, 62]
jsonList = json.dumps(alist)
print(jsonList)

#tuple to json string
atuple = (12, "Raghu")
jsonTuple = json.dumps(atuple)
print(jsonTuple)

#dictionary to json string
aDict = {"rollno":12, "name":"Raghu"}
jsonDict = json.dumps(aDict)
print(jsonDict)

#string to json
str = "Hello World"
jsonStr = json.dumps(str)
print(jsonStr)

#int to json
i = 25
jsonInt = json.dumps(i)
print(jsonInt)

#float to json
f = 25.256
jsonFloat = json.dumps(f)
print(jsonFloat)

#boolean to json
bool = True
jsonBool = json.dumps(bool)
print(jsonBool)

#null to json
a = None
jsonNone = json.dumps(a)
print(jsonNone)

```

Output

```

[12, 57, 41, 68, 47, 62]
[12, "Raghu"]
{"rollno": 12, "name": "Raghu"}
"Hello World"
25
25.256
true
null

```

Conclusion

In this [Python Tutorial](#), we learned how to parse JSON formatted string, convert some of the Python objects to JSON string and vice versa.

Python Programming

- Python Tutorial
- Install Python
- Install Anaconda Python
- Python HelloWorld Program
- Python Variables
- Python Variable Data Type Conversion
- Python Comments

Control Statements

- Python If
- Python If Else
- Python While Loop
- Python For Loop

Python String

- Python String Methods
- Python String Length
- Python String Replace
- Python Split String
- Python Count Occurrences of Sub-String
- Python Sort List of Strings

Functions

- Python Functions

Python Collections

- Python List
- Python Dictionary

Advanced

- Python Multithreading

Useful Resources

- Python Interview Questions