Cloud, APIs and Alerts > Introduction Python

JSON in Python

Now let's have a quick look at the JSON string format.

What is JSON

JSON is a method to format strings so that you can store different data types of data inside the string. This format is really inuitive and can be used by any program to encode data into a string so that you can send it to other programs.

If you want to encode a string with the following data:

```
Date: 23-05-2014
Time: 3:30pm
Location: Bangalore
```

You would do it in the following manner:

```
{
    "Date":"23-05-2014",
    "Time":"3:30pm",
    "Location":"Bangalore"
}
```

There the keywords “Date”, “Time” and “Locations” are called keys, and the values “23-05-2014”, “3:30pm” and “Bangalore” are called values.

##Why is it used?

You can think of JSON as a language for formatting strings. A language which is known world wide by almost all programmers. This is why people use the JSON format for the software API.

It helps programmer define what output their functions are going to give, without really defining what data is going to be present in the output.

Lets say, I have a function which checks the current time and return a string which contains the current time.

If the function was written in the normal method, the function would have to return an integer which depicts the current time. But what would happen if the function could not find out what the current time? Well it would respond back with a constant number, which the calling program would have to know from before.

Now if the function returned a json formatted string which contained the time, then it would return something of the sort:

```
{
    "Time":"4:45pm"
}
```

And if it was not able to retirve the time correctly it would return the following:

```
{
    "error":"Could not retireve the time"
}
```

As you can see, the calling funciton would only have to check if the JSON string had the key “Time” inside of it. If the string does not have the key “Time”, then the calling function would check for the error message which is the value associated with the “error” key.

Where is it used?

The JSON format is often used for serializing and transmitting structured data over a network connection. This means it is the perfect fit for online API that are made available by different companies.

The Bolt Cloud also gives API’s which return JSON formatted responses. You will learn more about API in the following chapters.

How to use JSON with Python

To use JSON string format with Python, you first need to import the JSON library using the following instruction

NOTE: The following instruciton have to be executed in the python console which you can launch using the command python3

```
import json
```

You can then create a JSON formatted string using the following instruction:

```
jsonString='{"Date":"23-05-2014",
"Time":"3:30pm",
"Location":"Bangalore"}'
```

To create a JSON object that can read the data from the string, use the following instructions

```
jsonObject=json.loads(jsonString)
```

You can then view the values present in the JSON object using the following instruciton

```
jsonObject["Date"]
jsonObject["Time"]
jsonObject["Location"]
```

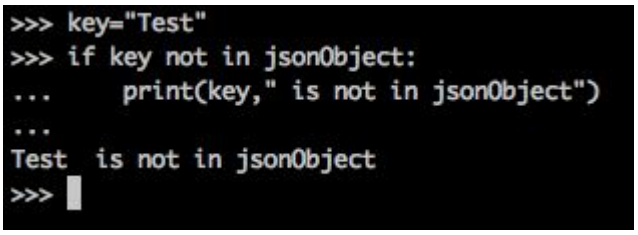
The python console will show you the data in the following manner:

```
Type "help", "copyright", "credits" or "license" for more information.
>>> import json
>>> jsonString='{"Date":"23-05-2014","Time":"3:30pm","Location":"Bangalore"}'
>>> jsonObject=json.loads(jsonString)
>>> jsonObject["Date"]
'23-05-2014'
>>> jsonObject["Time"]
'3:30pm'
>>> jsonObject["Location"]
'Bangalore'
```

To check if a certain key exists in the JSON object, you can use the following instructions


```
key="Test"
if key not in jsonObject:
    print(key," is not present in the jsonObject")
```

The python console will show you the data in the following manner:



To create a fresh JSON object, you can use the following instruction

```
freshJsonObject=json.loads("{}")
```

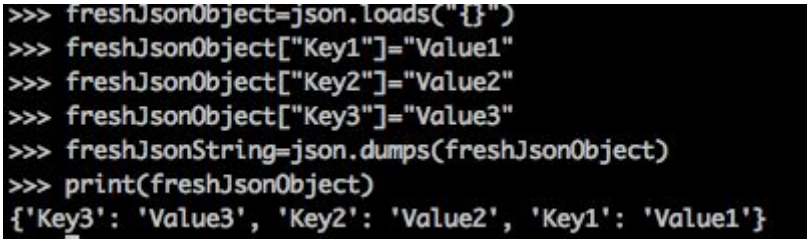
You can then add new keys to this JSON object, by using the following instructions.

```
freshJsonObject["Key1"]="Value1"
freshJsonObject["Key2"]="Value2"
freshJsonObject["Key3"]="Value3"
```

To convert this JSON object into a JSON formatted string use the following instructions

```
freshJsonString=json.dumps(freshJsonObject)
print(freshJsonObject)
```

The following image displays what the python console will display when you run these instructions.



In this section, we learned what JSON is, why it is used, where it is used and how to use JSON with python.

