

# Python One Cool Cat (Meow Facts) Tkinter Tutorial

## Contents

Python One Cool Cat (Meow Facts) Tkinter Tutorial.....	1
Public API's .....	1
Meow Facts API.....	2
What is JSON? .....	2
What is a Dictionary? .....	2
Dictionary Basics .....	4
Tutorial: Meow Facts .....	4
Install the Python Requests Library.....	4
Developing a Web API Program.....	4
Time for Meow Facts GUI .....	6
A Good Icon.....	6
The Code .....	6
get_data().....	8
create_widgets.....	9
Assignment Submission.....	9

Time required: 60 minutes

- Comment each line of code as show in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

## Public API's

**NOTE:** You don't have to use the API in this tutorial, you can use any API you wish.

There are hundreds of public API's available. This website contains a list of some of them.

<https://github.com/public-apis/public-apis>

## Meow Facts API

Meow Facts is a simple API with only item returned, a random Meow Fact.

We are going to create a Python program to get a Meow Fact and display it.

Before we do that, we have a few new concepts to go through.

---

### What is JSON?

The Meow Facts is a free https-based joke API which provides a JSON response.

Raw API data is typically in a text file in JSON or XML format. For our API projects, we will use JSON.

JavaScript Object Notation (JSON) is an open standard file and data interchange format. It uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types.

Go to <https://meowfacts.herokuapp.com/> to see some raw JSON data in two different formats.

This JSON is a list inside of a dictionary. You work from the outside in.

1. Access the dictionary key: ("data")
2. Access the list inside the dictionary: ("data")[0]

```
meow_facts.get("data")[0]
```

---

### What is a Dictionary?

A dictionary is a list of items. Dictionaries are mutable, they can be changed.

Here is a Python list that contains the number of days in the months of the year.

```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

If we want the number of days in January, use `days[0]`. December is `days[11]` or `days[-1]`.

Here is a dictionary of the days in the months of the year:

```
days = {'January':31, 'February':28, 'March':31, 'April':30, 'May':31,
        'June':30, 'July':31, 'August':31, 'September':30, 'October':31,
        'November':30, 'December':31}
```

To get the number of days in January, we use `days['January']`.

One benefit of using dictionaries is the code is more readable. We don't have to figure out which index in the list a given month is at.

Unlike other sequences, items in dictionaries are unordered. The first item in a list named `spam` would be `spam[0]`. There is no "first" item in a dictionary. The dictionary does remember the order of entry. While the order of items matters for determining whether two lists or tuples are the same, it does not matter in what order the key-value pairs are typed in a dictionary.

---

## Dictionary Basics

Parameter	Details
key	The desired key to lookup
value	The value to set or return

## Tutorial: Meow Facts

---

### Install the Python Requests Library

To retrieve information from API's, we expand the functionality of default Python. Requests is a Python Library that lets you send HTTP/1.1 requests, add headers, form data, multipart files, and parameters with Python dictionaries. It lets you access the response data in the same way.

Requests are synchronous. Only one HTTP call can be made at a time.

In Windows, go to a command prompt.

```
pip install requests
```

The command should either install requests or confirm that it is already installed.

### Developing a Web API Program

A good way to start developing a Python Web API program is to keep it simple until you can get it to work. After you have figured out how to access the json data, time to put that into a program.

Create the following program named **meow\_facts.py** which demonstrates how to access a Web API data with Python.

Right Click the URL below, Click Copy Link location. Paste this into your program to avoid typo's.

<https://meowfacts.herokuapp.com/>

```

1  """
2      Name: meowfacts_cli.py
3      Author: William A Loring
4      Created: 04/18/21
5      Purpose: Get random cat facts from https://meowfacts.herokuapp.com
6  """
7
8  # Install requests module
9  # pip install requests
10 # Import requests module
11 import requests
12
13 # URL for mewfacts API
14 URL = "https://meowfacts.herokuapp.com/"
15
16 # Get requests object from URL
17 response = requests.get(URL)
18 # Print response for troubleshooting
19 # Comment out this line when the program is working
20 print(response)
21
22 # Convert requests json requests object to Python dictionary
23 meowfacts = response.json()
24
25 # Comment out this line when the program is working
26 print()
27 # Print dictionary for troubleshooting
28 # Comment out this line when the program is working
29 print(meowfacts)
30
31 # Print just the data using dictionary created from API Json
32 print()
33 print(meowfacts.get("data")[0])

```

Example run:

```

<Response [200]>

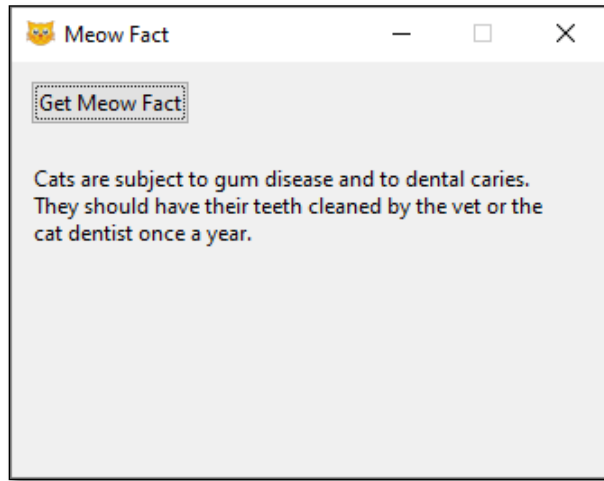
{'data': ["Siamese kittens are born white because of the heat inside the mother's uterus before birth. This heat keeps the kittens' hair from darkening on the points."]}

Siamese kittens are born white because of the heat inside the mother's uterus before birth. This heat keeps the kittens' hair from darkening on the points.

```

This is a very simple example of an API call.

## Time for Meow Facts GUI



Let's jazz up our program with a little Object Oriented TKinter.

---

### A Good Icon

The most important part of creating a GUI program is getting just the right icon for the program. Icon Archive has a lot of open-source icon (ico) files you can use in your program.

<https://iconarchive.com/>

### The Code

Once you have a good framework of a program established, it is easy to change it to another API.

```

1  """
2      Name: meowfacts_gui.py
3      Author: William Loring
4      Created: 05/01/2021
5      Purpose: Get a random meowfact
6  """
7  from tkinter import *
8  from tkinter.ttk import *
9  import requests
10
11
12  class MeowFactsGUI:
13      # URL API endpoint as a class constant
14      # It belongs to the class, not the object
15      URL = "https://meowfacts.herokuapp.com"
16
17      def __init__(self):
18          # Set this to True for troubleshooting
19          self.IS_DEBUGGING = False
20
21          # Create Tkinter GUI
22          self.root = Tk()
23          self.root.title("Meow Fact")
24          self.root.geometry("330x230+250+250")
25          self.root.resizable(0, 0)
26          self.root.iconbitmap("cat2.ico")
27
28          self.create_widgets()
29          mainloop()

```

## get\_data()

```
31 # ----- GET AND DISPLAY API DATA -----#
32 def get_data(self, *args):
33     # Use the requests.get() function to get API data
34     response = requests.get(MeowFactsGUI.URL)
35     # Watch for exception
36     response.raise_for_status()
37
38     # If the status_code is 200, successful connection
39     if (response.status_code == 200):
40
41         # Convert the JSON data into a Python dictionary
42         self.data = response.json()
43
44         # Used to debug process
45         if (self.IS_DEBUGGING == True):
46
47             # Display the status code
48             print(
49                 f"\nStatus code: {response.status_code} \n")
50
51             # Display the raw JSON data from the API
52             print("Raw API data: ")
53             print(response.text)
54
55             # Display the Python dictionary
56             print("\nJSON data converted to a Python dictionary:")
57             print(self.data)
58
59             self.lbl_display.configure(
60                 text=self.data.get("data")[0])
61
62         else:
63             self.lbl_display.config(
64                 text=f"API Unavailable {response.status_code}")
65
```



---

## create\_widgets

```
66 # ----- CREATE WIDGETS -----#
67 def create_widgets(self):
68     # Set the wraplength property to wrap the text at 300 pixels
69     self.lbl_display = Label(wraplength=300, justify="left")
70
71     self.btn_get_data = Button(
72         text="Get Meow Fact",
73         command=self.get_data
74     )
75
76     self.btn_get_data.grid(row=0, column=0, sticky=W)
77     self.lbl_display.grid(row=1, column=0, sticky=W)
78
79     # Set padding for all widgets
80     for child in self.root.winfo_children():
81         child.grid_configure(padx=10, pady=10)
82
83     # Bind both enter key to the get_data method
84     # When either Enter key is pressed,
85     # the get_data method will be fired
86     # <Return> - Enter key on main keyboard
87     # <KP_Enter> - Enter key on number pad/key pad
88     self.root.bind("<Return>", self.get_data)
89     self.root.bind("<KP_Enter>", self.get_data)
90
91
92 # Create program object to start program
93 meowfacts_gui = MeowFactsGUI()
```

You can use this program as a template to develop your own API programs.

---

## Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.