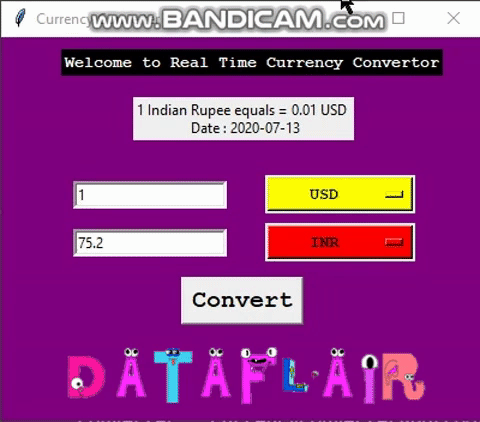
Python Project on Currency Converter

**About the Python Project**



( this gif may not work in Microsoft word.)

In this Python project idea, we are going to build an exciting project through which you can convert currencies. For a user interface, we are going to use the tkinter library.

### You can get the source code here,

<https://github.com/chikii/Projects/tree/master/Currency%20Converter>

**Prerequisites**

### The project in Python requires you to have basic knowledge of python programming and the pygame library.

Libraries ->

tkinter - For User Interface (UI)

requests - to get url

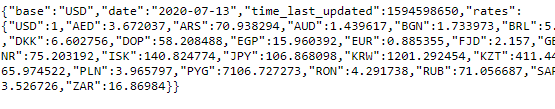
To install the tkinter and requests library, type the following code in your terminal.

|  |
| --- |
| pip install tkinter  pip install requests |

## 

## **Steps to Build the Python Project on Currency Converter**

1. Real time Exchange rates
2. Import required Libraries
3. CurrencyConverter Class
4. UI for CurrencyConverter
5. Main Function
6. To get real time exchange rates, we will use <https://api.exchangerate-api.com/v4/latest/USD>



Here, You can see we have JSON data in which,

**Base : USD**

It means we have our base currency is USD. which means to convert any currency we have to first convert it into USD then from USD, we will convert it in whichever currency we want.

**Date and time :** it shows the last updated date and time.

**Rates :** It is the exchange rate of currencies with base currency USD.

1. **Import the libraries :**

For this project based on Python, we are using the tkinter and requests library. So we need to import the library.

|  |
| --- |
| import requests  from tkinter import \* |

1. **Create the CurrencyConverter class :**

Now we will create the CurrencyConverter class which will get the real time exchange rate and convert the currency and return the converted amount.

1. Let’s create the **constructor of class.**

|  |
| --- |
| class CurrencyConverter():  def \_\_init\_\_(self,url):  self.data= requests.get(url).json()  self.currencies = self.data['rates'] |

requests.get(url) load the page in our python program and then .json() will convert the page into the json file. We store it in a data variable.

1. **Convert()** method :

|  |
| --- |
| def convert(self, from\_currency, to\_currency, amount):  initial\_amount = amount  #first convert it into USD if it is not in USD.  # because our base currency is USD  if from\_currency != 'USD' :  amount = amount / self.currencies[from\_currency]    # limiting the precision to 2 decimal places  amount = round(amount \* self.currencies[to\_currency], 2)  return amount |

This method takes

*From\_currency* - currency from which you want to convert.

*to \_curremcy* - currency in which you want to convert.

*Amount* - how much amount you want to convert.

Return the converted amount.

**Example ->**

|  |
| --- |
| url = 'https://api.exchangerate-api.com/v4/latest/USD'  converter = CurrencyConverter(url)  print(converter.convert('INR','USD',100)) |

**OUTPUT - 1.33**

100 Indian rupees = 1.33 US dollars

As simple as that.

**4. Now let’s create a UI for our Converter.**

To Create UI we will make a **class CurrencyConverterUI**

|  |
| --- |
| class CurrencyConverterUI():  def \_\_init\_\_(self,root,converter):  self.root =root  root.title = 'Currency Converter'  self.converter = converter |

*Root* - Root is out tkinter main Frame where we will build UI for Converter.

*Converter -* Currency Converter object which we will use to convert currencies.

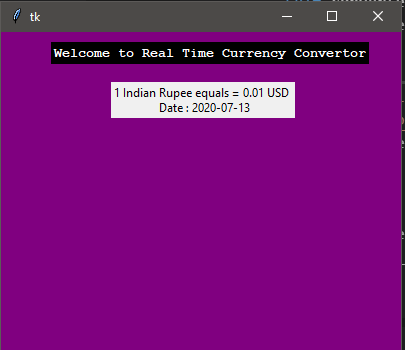
Let’s Create the **Converter**

|  |
| --- |
| def create\_converter(self):  # Set the background colour of GUI window  root.configure(background = 'purple')  # Set the configuration of GUI window (WidthxHeight)  root.geometry("400x320")  # Create welcome to Real Time Currency Convertor label  headlabel = Label(root, text = 'Welcome to Real Time Currency Convertor', fg = 'white', bg = "black")  headlabel.config(font = ('Courier',10,'bold'))  # Create a 'DESC' label  label\_desc = Label(root, text = f"1 Indian Rupee equals = {self.converter.convert('INR','USD',1)} USD \n Date : {self.converter.data['date']}")  # Placing on screen  headlabel.place(x=50, y=10)  label\_desc.place(x = 110, y= 50) |

**Note -** This function is part of CurrencyConverterUI class.

First we set up the frame and add some info in it.

After the execution of this part of code our frame looks like something.



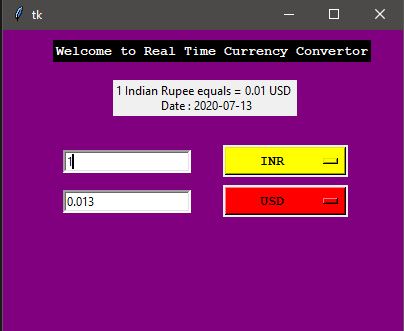
Now let’s create the entry box for the amount and options of currency in the frame. So That users can enter the amount and choose among currencies.

|  |
| --- |
| # Create Entry box  self.Amount = Entry(root,bd = 3)  self.converted\_amount\_field = Entry(root,bd=3)  self.Amount.insert(0, 1)  self.converted\_amount\_field.insert(0,0.013)  # Create a dropdown  self.variable1 = StringVar(root)  self.variable2 = StringVar(root)    # initialise the variables with default values  self.variable1.set("INR")  self.variable2.set("USD")  from\_curr\_options = OptionMenu(root, self.variable1, \*converter.currencies.keys())  from\_curr\_options.config(width=10, font=('Courier', 10,'bold'), bg = 'yellow', fg = 'black')  from\_curr\_options.pack()  to\_curr\_options = OptionMenu(root, self.variable2, \*converter.currencies.keys())  to\_curr\_options.config(width=10, font=('Courier', 10,'bold'), bg = 'red', fg = 'black')  to\_curr\_options.pack()  # Placing on screen  self.Amount.place(x = 60, y= 120)  self.converted\_amount\_field.place(x = 60,y = 160)  from\_curr\_options.place(x = 220 , y = 115)  to\_curr\_options.place(x = 220 , y = 155) |

**NOTE -** this code is part of create\_converter function

After the successful Execution of code till now.

Our Frame will look like something.



Now Let’s add the **CONVERT button** which will call the perform function.

|  |
| --- |
| self.button1 = Button(root, text = "Convert", fg = "black", command = self.perform)  self.button1.config(font=('Courier', 15, 'bold'))  self.button1.place(x = 150, y = 200) |

*Command = self.perform* - It means on click it will call perform().

**perform() method :**

Perform method will take the user input and convert the amount into desired currency and display it on converted\_amount entry box.

|  |
| --- |
| def perform(self,):  amount = float(self.Amount.get())  from\_curr = self.variable1.get()  to\_curr = self.variable2.get()  converted\_amount = self.converter.convert(from\_curr,to\_curr,amount)  converted\_amount = round(converted\_amount, 2)  self.converted\_amount\_field.delete(0,END)  self.converted\_amount\_field.insert(0,converted\_amount) |

**5. Let’s create the main function.**

First, we will create the Converter.

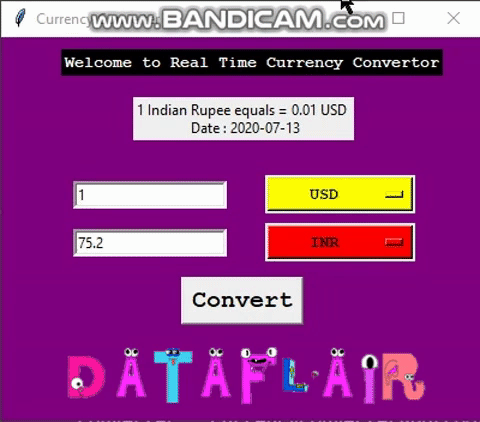
Second, Create the Frame

Third, Create the UI for Converter

Fourth, Buid Converter

|  |
| --- |
| if \_\_name\_\_ == '\_\_main\_\_':  url = 'https://api.exchangerate-api.com/v4/latest/USD'  converter = CurrencyConverter(url)  root = Tk()    Converter = CurrencyConverterUI(root,converter)  Converter.create\_converter()  root.mainloop() |

**OUTPUT :**



## Summary

In this article, you worked on the Python project to build your own Currency Converter.

I hope you got to learn new things and enjoyed building this interesting Python project. Do share the article on social media with your friends and colleagues.