Flask – Variable Rule

Last Updated : 05 Jan, 2023

microweb framework written in Python it is classified as a micro framework because it doesn't require particular tools or libraries, and it has no database abstraction layer form validation or any other components where pre-existing third-party libraries provide common functions it is designed to make getting started quick and easy with the ability to scale up complex application.

Flask -Variable Rules:

- By using variable rules we can create a dynamic URL by adding variable parts, to the rule parameter.
- We can define variable rule by <variable-name> using this syntax in our code.
- Variable is always passed as a keyword argument to the function with which the rule is properly associated.

Dynamic URLs Variable In Flask

The following converters are available in Flask-Variable Rules:

- **String** – It accepts any text without a slash(the default).
- **int** – accepts only integers. ex =23
- **float** – like int but for floating point values ex. = 23.9
- **path** – like the default but also accepts slashes.
- **any** – matches one of the items provided.
- **UUID** = accepts UUID strings.

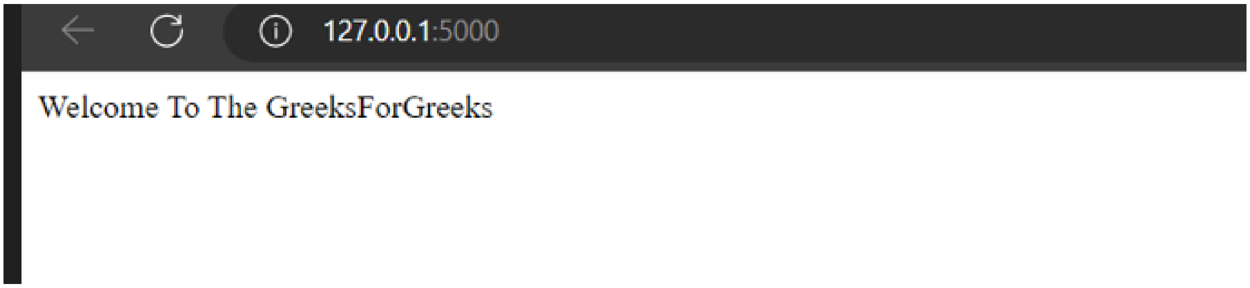
Simple flask program

In this code, we import the first Flask and then we initialize the Flask function and then we return the simple welcome line we need to run the Flask run command in the terminal to execute our code.

Python3

```python
# first we import flask
from flask import Flask

# Initialize flask function
app = Flask(__name__)


@app.route('/')
def msg():
    return "Welcome To The GreeksForGreeks"

 # we run code in debug mode
 app.run(debug=True)
```

Output:



```
Welcome To The GreeksForGreeks
```

String Variable in Flask

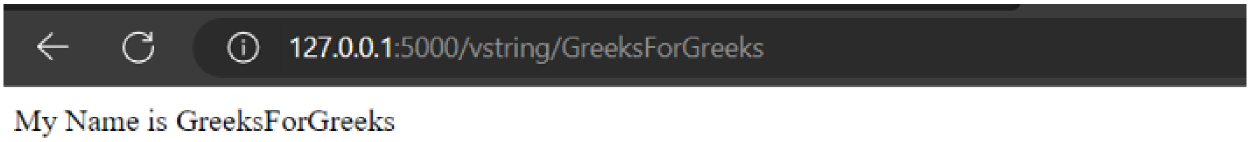
In this code, we import the first Flask and then we initialize the Flask function after that we made a string function string is already defined in the code we don’t need to specify the string and then we execute our code by a run in the terminal Flask run. and in URL we need to write **127.0.0.1:5000/vstring/<name>** to run our vstring function.

Python3

```python
# First we import flask
from flask import Flask
```

```
# initialize flask
app = Flask(__name__)

# Display first simple welcome msg
@app.route('/')
def msg():
    return "Welcome"

# We defined string  function
@app.route('/vstring/<name>')
def string(name):
    return "My Name is %s" % name

# we run app debugging mode
app.run(debug=True)
```

Output :



```
127.0.0.1:5000/vstring/GreeksForGreeks
```

```
My Name is GreeksForGreeks
```

Integer Variable in Flask

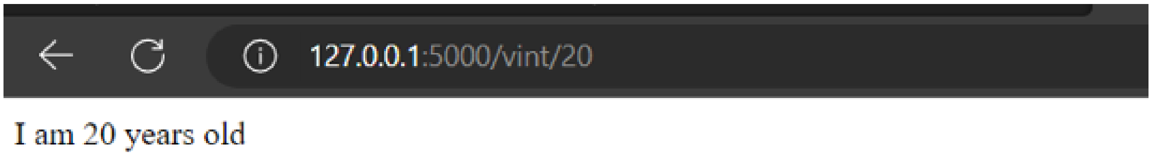
In this code, we import the first Flask, and then we initialize the Flask function after that we made an int function, int is not defined in a Flask so we need to first define it then we return the int function and run Flask run in terminal and for int page, we need to write function name in URL and also the integer value.

Python3

```
# first we import flask
from flask import Flask

# Initialize flask function
```

```
app = Flask(__name__)

@app.route('/')
def msg():
    return "Welcome"

# define int function
@app.route('/vint/<int:age>')
def vint(age):
    return "I am %d years old " % age

# we run our code in debug mode
app.run(debug=True)
```

Output:



127.0.0.1:5000/vint/20

I am 20 years old

Float Variable in Flask

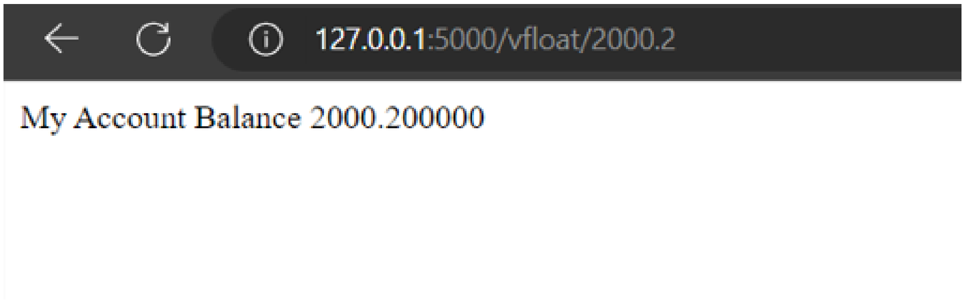
In this code, we import the first Flask and then we initialize the Flask function after that we made a float function float is not defined in a Flask so we need to first define it then return the float function and run the Flask run in the terminal and for float page, we need to write function name in URL and also the floating value.

Python3

```
# first we import flask
from flask import Flask

# Initialize flask function
app = Flask(__name__)
```

```
@app.route('/')
def msg():
    return "Welcome"

# define float function
@app.route('/vfloat/<float:balance>')
def vfloat(balance):
    return "My Account Balance %f" % balance

 # we run our code in debugging mode
app.run(debug=True)
```

Output:



127.0.0.1:5000/vfloat/2000.2

My Account Balance 2000.200000

prath... 3

Previous Article

Flask - HTTP Method

Next Article

Redirecting to URL in Flask

Similar Reads

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses...

4 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to...

4 min read

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is know...

8 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would...

12 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used f...

11 min read

Using Request Args for a Variable URL in Flask

This article will teach us how to use Request Arguments in our Flask application. First, we will understand. What are the Request Arguments? What are the...

5 min read

Python Flask - ImmutableMultiDict

MultiDict is a sub-class of Dictionary that can contain multiple values for the same key, unlike normal Dictionaries. It is used because some form elements ha...

2 min read

Handling 404 Error in Flask

Prerequisite: Creating simple application in Flask A 404 Error is showed whenever a page is not found. Maybe the owner changed its URL and forgot to...

4 min read

Python | Using for loop in Flask

Prerequisite: HTML Basics, Python Basics, Flask It is not possible to write front-end course every time user make changes in his/her profile. We use a template...

3 min read

How to Build a Simple Android App with Flask Backend?

Flask is an API of Python that allows us to build up web applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's...

8 min read

Article Tags : Python Technical Scripter Technical Scripter 2022

Practice Tags : python

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us

Legal

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation