

آموزش flask

جلسه دوم

نویسنده: بهنام حسن بیگی

Mlops fanapsoft /AIFA TEAM

در این جلسه مباحث اصلی یا base فلسک رو توضیح میدم:

به نظر خودم بهترین راه برای یادگیری اینه که هر جلسه شما یک پوشه خاص برای اون قسمت بسازید چیزی شبیه به کاری که من انجام دادم و این مراحل رو انجام بدید (در طول آموزش این مراحل ثابت هستند)

۱ پوشه ی برای نوشتن کد ها تون ایجاد کنید

PC > Desktop > flask > learn-flask

| Name | Date modified | Type | Size |
|----------------------|---------------------|--------------------|------|
| .git | 10/25/2020 12:34 AM | File folder | |
| 0-1 flask-helloworld | 10/25/2020 12:34 AM | File folder | |
| 0-2 flask-basic | 10/25/2020 2:58 AM | File folder | |
| README.md | 10/25/2020 12:32 AM | Markdown Source... | 1 KB |

مرحله دوم یک virtual env تو همون پوشه ی که ساختی تشکیل بده و فعالش کن

```
C:\Desktop\flask\learn-flask\0-2 flask-basic> python -m venv behnam
```

Python -m venv ENV_NAME

بعدش اکتیو کردن

```
PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> .\behnam\Scripts\activate
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
```

مرحله سوم نصب پکیج flask

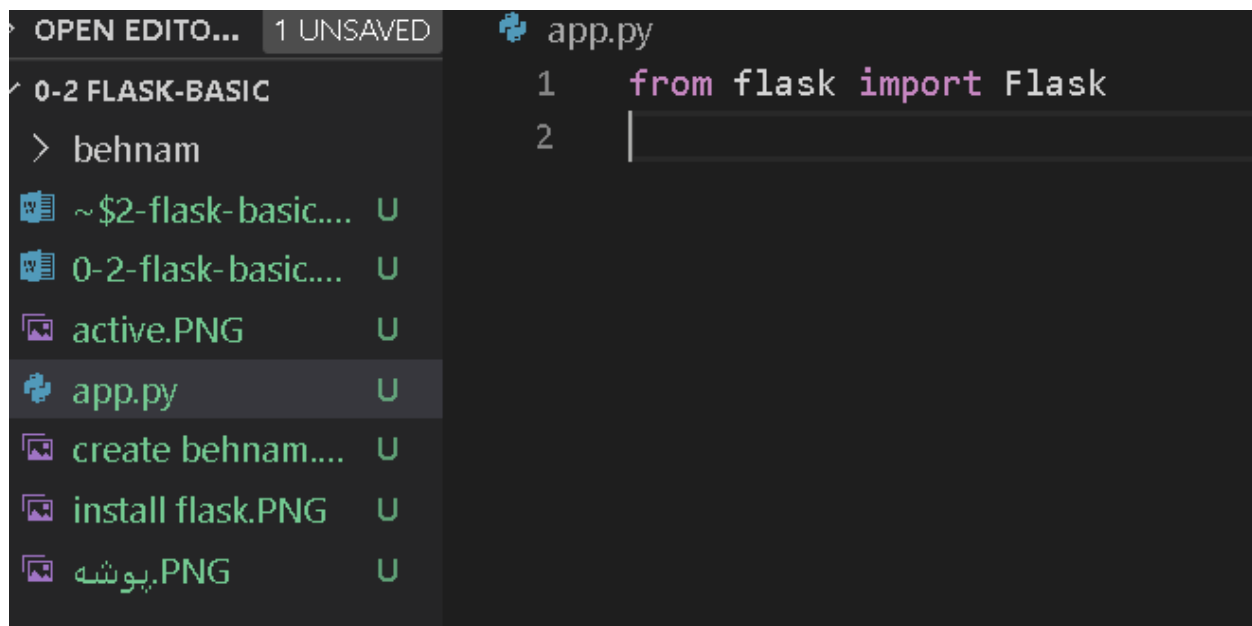
```
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> pip install flask
Collecting flask
  Using cached Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click>=5.1
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting itsdangerous>=0.24
  Using cached itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Using cached Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug>=0.15
  Using cached Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting MarkupSafe>=0.23
  Using cached MarkupSafe-1.1.1-cp38-cp38-win_amd64.whl (16 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, flask
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsda
WARNING: You are using pip version 20.2.1; however, version 20.2.4 is available.
```

در طی آموزش flask اکثر مراحل بالا ثابت مگر اینکه پکیجی رو بخواهیم نصب کنیم.

خب الان می خواهیم راجب هر قسمتی رو که تو جلسه قبل نوشتیم صحبت کنیم:

در ابتدای کد ما از ماژول flask کلاس Flask رو import کردیم که در کدمون از اون استفاده کنیم

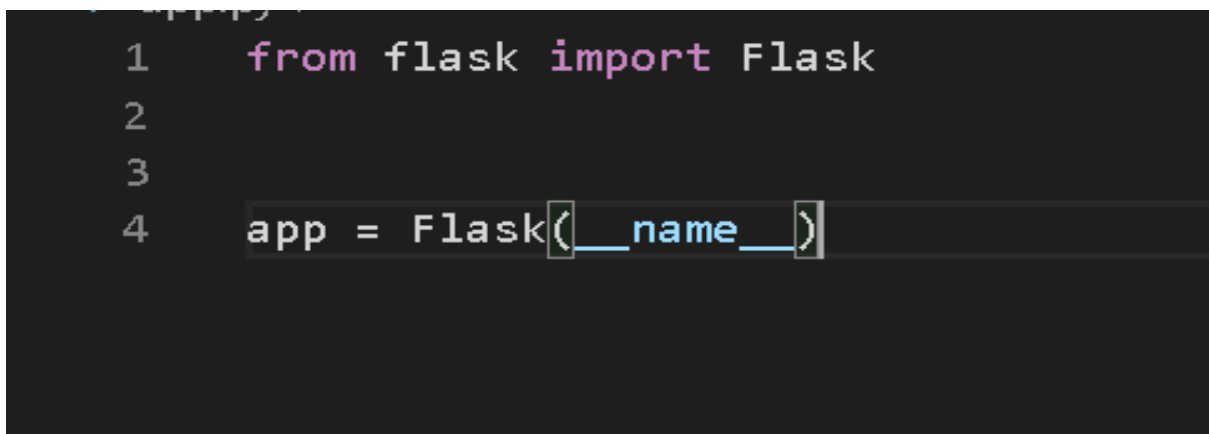
```
from flask import Flask
```



```
OPEN EDITO... 1 UNSAVED
0-2 FLASK-BASIC
  > behnam
    ~$2-flask-basic.... U
    0-2-flask-basic.... U
    active.PNG U
    app.py U
    create behnam.... U
    install flask.PNG U
    پوشه.PNG U

1 from flask import Flask
2
```

تو خط چهارم یک متغیر تعریف میکنیم به اسم app و یک object ساختیم از Flask



```
1 from flask import Flask
2
3
4 app = Flask(__name__)
```

این app object اصلی کد ما هست حالا در ادامه بیش تر باهش آشنا میشیم و خیلی باهش کار داریم فقط در این حد بدونید که تمام کار web-application شما با کمک این app انجام میشه.

Class فلسک یه سری پارامتر به عنوان ورودی میگیره (که در اینجا همون __name__)

حالا در interpreter powershell پایتون رو اجرا میکنم تا بیش با __name__ آشنا بشید(در همون virtualenv که تشکیل دادید انجام بدید)

```
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> python
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from flask import Flask
>>> help(Flask)
Help on class Flask in module flask.app:

class Flask(flask.helpers._PackageBoundObject)
|   Flask(import_name, static_url_path=None, static_folder='static', static_host=None, host_matching=False,
|   template_folder='templates', instance_path=None, instance_relative_config=False, root_path=None)
|
|   The flask object implements a WSGI application and acts as the central
|   object. It is passed the name of the module or package of the
|   application. Once it is created it will act as a central registry for
|   the view functions, the URL rules, template configuration and much more.
|
|   The name of the package is used to resolve resources from inside the
|   package or the folder the module is contained in depending on if the
|   package parameter resolves to an actual python package (a folder with
|   an :file: __init__.py file inside) or a standard module (just a .py file).
|
|   For more information about resource loading, see :func:open_resource.
|
|   Usually you create a :class:Flask instance in your main module or
```

در ابتدا python رو اجرا کرده سپس کلاس flask رو فراخوانی کنید

Python

from flask import Flask

سپس با کمک تابع help میتونید امکانات و کلاپارامتر های که کلاس Flask میگیره مشاهده کنید

help(Flask)

این تابع به شما کمک میکنه که از کلاس های که دارید تو کدتون استفاده میکنید اطلاعات خوبی به دست بیارید

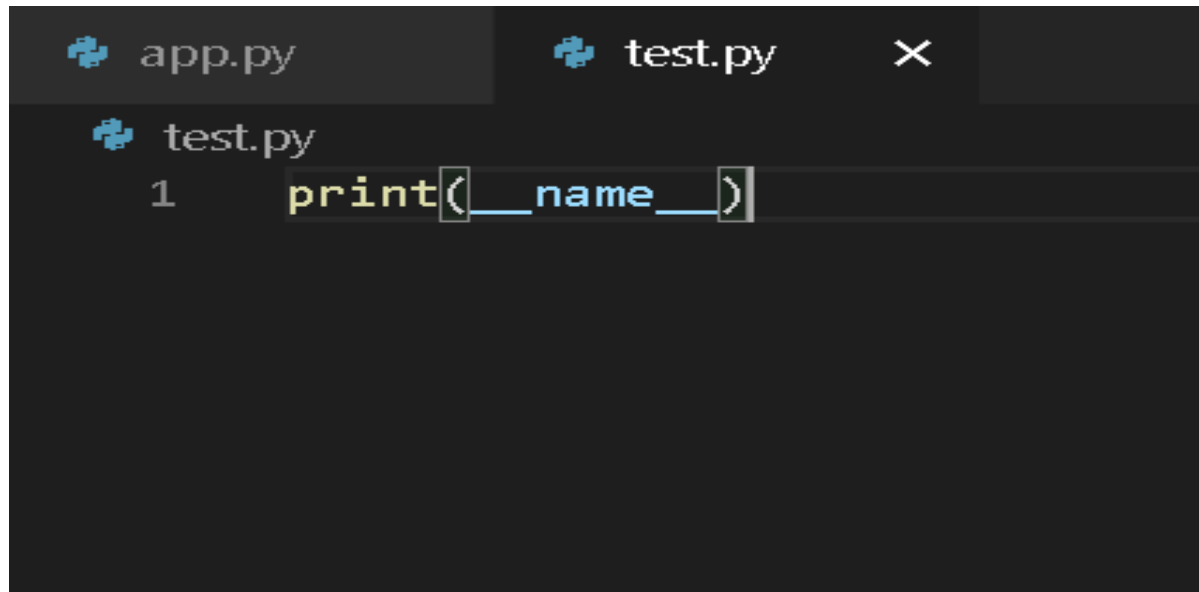
پارامتری یا argument که ما به کلاس Flask مون دادیم `__name__` که به معنای dunder name که این کلمه از

Double underline گرفته شده و ابتدای هر کلمه رو به هم چسبانندن و کلمه ی dunder اختراع شده.

یه سری از پارامتری خاص در پایتون وجود داره که ابتدا و انتهایشون از دو تا خط تیره تشکیل شده که به اون ها هم dunder میگوییم.

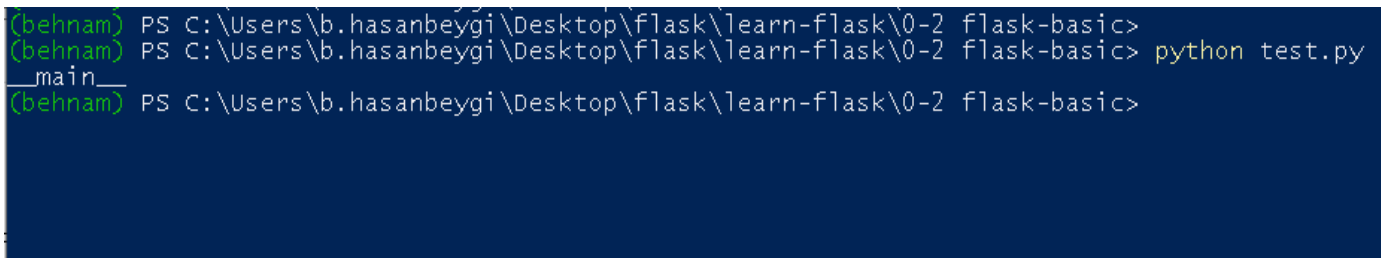
متغیر `__name__` یک متغیر از پیش تعریف شده س که می اد به اسم اون کدی که نوشتیم و میخواد در جای دیگه import بشه اشاره می کنه. به طور مثال کد رو نوشتیم و می تونیم اون رو در interpreter پایتون import کنیم و کلاس های که نوشتیم استفاده کنیم

به طور مثال یک فایل میسازیم به اسم test.py



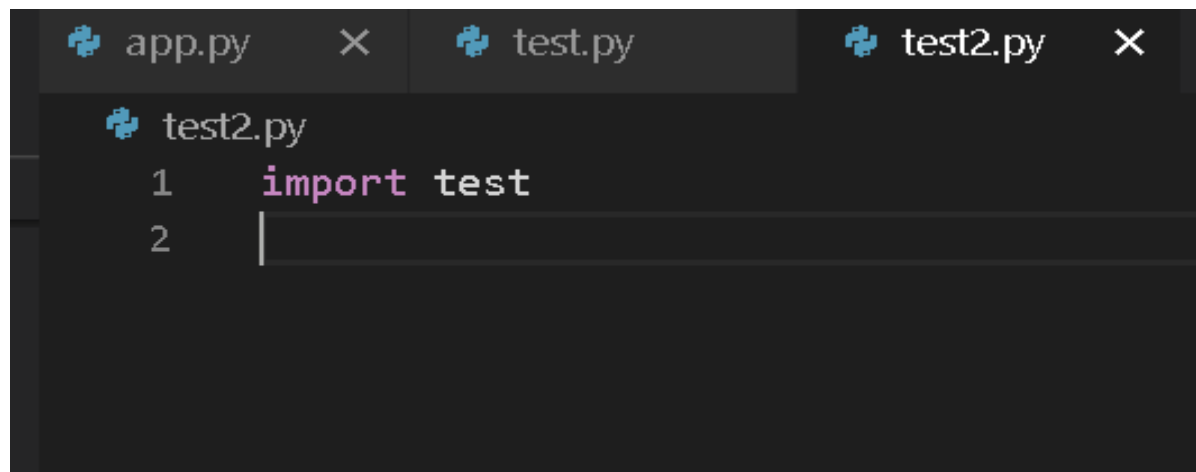
```
app.py test.py X
test.py
1 print(__name__)
```

و بعد اگر اون رو اجرا کنیم باید __main__ را به ما نشان بدهد



```
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> python test.py
__main__
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
```

حالا بیاید یک فایل جدید بسازید به اسم test2.py و فایل test رو import کنید و سپس کد را اجرا کنید همان طور که میبینید اسم test اجرا می شود



```
app.py X test.py test2.py X
test2.py
1 import test
2 |
```

```

main
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> python test2.py
test
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>

```

زمانی که یک فایل رو به صورت مستقیم اجرا می کنید `dunder name` اون فایل همیشه `__main__` و اگر در جای اون رو `import` کنید `dunder name` اون فایل همیشه اون فایلی که `import` کردید

قسمت بعدی کد مربوط به خط 7 که از یک `decorator` استفاده کردم و در خط 8 یک فانکشن تعریف کردم برای ساخت یک `view`

```

app.py > index
1  from flask import Flask
2
3
4  app = Flask(__name__)
5
6
7  @app.route('/')
8  def index():
9      return "hello world"

```

به طور کلی همه ی سایت ها از `path` های تشکیل شده ان که مسیر دلخواه شما رو برای رسیدن به صفحه ی مورد نظرتون مشخص میکنن مثلا شما برای ثبت نام در سایتی به ادرس <http://nameofsite.com/signup> میرید که یک `path` یا مسیره برای شما یا <http://nameofsite.com/login> که در اینجا لاگین میکنید

با `@app.route .. decorator` ما مسیری که میخواییم بهش دسترسی پیدا رو مشخص میکنیم که اینجا `<= /`

در زیر این دکراتور ما میاییم یه تابع تعریف میکنیم که میخواییم محتویات اون سایت رو به ما نشون بده.

به طور مثال در زیر اون تابع یا فانکشنی که نوشتید میتونید این تست رو انجام بدید:

```

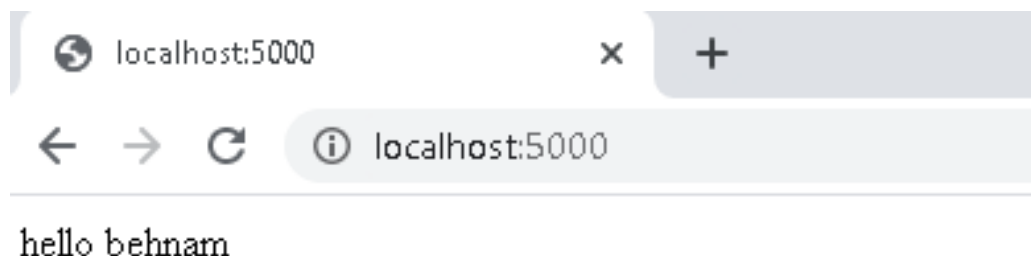
1  from flask import Flask
2
3
4  app = Flask(__name__)
5
6
7  @app.route('/')
8  def index():
9      name = "behnam"
10     return f"hello {name}"

```

بعد با command

flask run رو اجرا کنید و نتیجه رو ببینید.

اون f پشت متنی که نوشتیم استفاده از f string هاست که بعدا راجبش خیلی صحبت میکنیم



حالا میاییم route که نوشتیم رو تغییر میدیم که ببینم حاصل کار چیه:

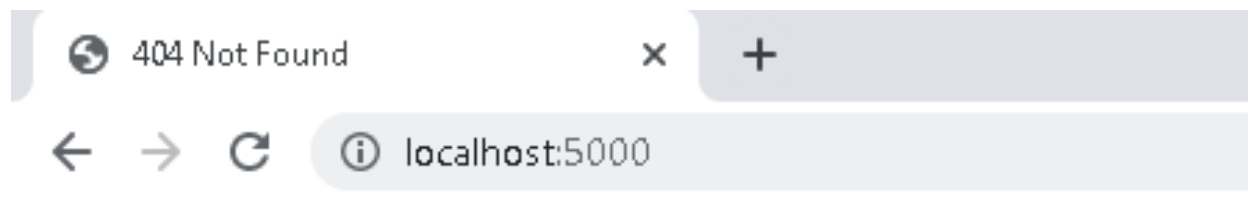
Route رو به /home تغییر دادیم

```
app.py / index
1   from flask import Flask
2
3
4   app = Flask(__name__)
5
6
7   @app.route('/home')
8   def index():
9       name = "behnam"
10      return f"hello {name}"
```

در این مرحله باز با command

flask run وب اپلیکیشن رو اجرا میکنیم

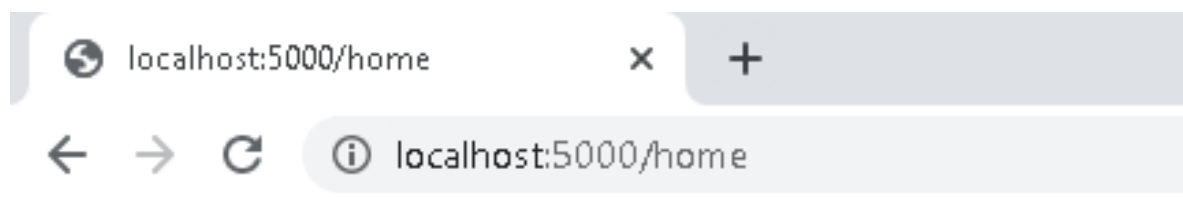
اگر در مرورگر localhost:5000 را باز کنید با خطای NOT FOUND مواجه می شوید که دقیقا درسته!!! شما route برای / درست نکردید که اون رو بتونید ببینید:



Not Found

The requested URL was not found on the server. If you entered the URL,

بعد به این ادرس برید <http://localhost:5000/home> و میتونید view که نوشتید رو مشاهده کنید.

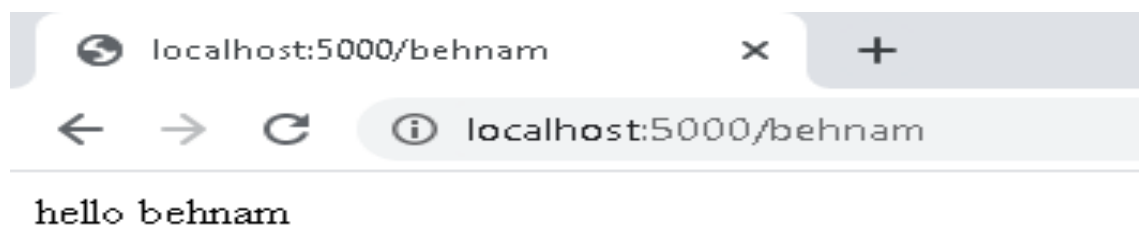


hello behnam

یک فانکشن یا تابع میتونه چندین route رو بگیره به عنوان مثال:

```
app.py > index
1  from flask import Flask
2
3
4  app = Flask(__name__)
5
6
7  @app.route('/home')
8  @app.route('/behnam')
9  def index():
10     name = "behnam"
11     return f"hello {name}"
```

یک route به اسم /behnam تشکیل دادیم که اگر اپ رو اجرا کنید و تو مرورگرتون چک کنید /home و /behnam صفحه ی شبیه به هم هستن:



حالا میخوایم درباره تابع ها صحبت کنیم. نمیتوان دو تابع هم اسم برای دو route مختلف نوشت یعنی هر route باید یک تابع با اسم خاص داشته باشد به عنوان مثال برای route های که در بالا نوشته شد

اسم تابع `index()` بود اگر در `route` بعدی اسم `index()` را باز تکرار کنیم با چنین خطای رو به رو می شویم:

```
app.py > index
1  from flask import Flask
2
3
4  app = Flask(__name__)
5
6
7  @app.route('/home')
8  @app.route('/behnam')
9  def index():
10     name = "behnam"
11     return f"hello {name}"
12
13  @app.route('/behnam2')
14  def index():
15     return "hello from behnam2"
```

حالا اگر برنامه رو اجرا کنیم با چنین خطای رپ به رو می شیم `flask run` را زده نگاه میکنیم:

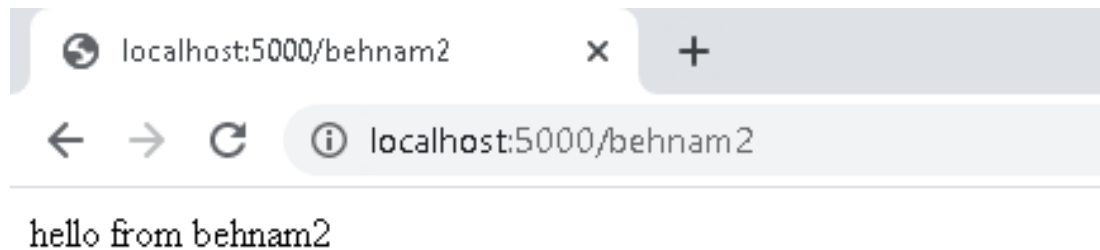
```
In wrapper_func
return f(self, *args, **kwargs)
File "c:\users\b.hasanbeygi\desktop\flask\learn-flask\0-2 flask-basic\behnam\lib\site-packages\
2, in add_url_rule
raise AssertionError(
AssertionError: View function mapping is overwriting an existing endpoint function: index
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
```

به این معناس که دو بار فانکشن یا تابع در کد تکرار شده و کاملاً خطا مشخص و واضح نشون داده شده.

حالا اگه این فانکشن دومی رو از `index()` به `index2()` تغییر بدم و یا یک اسم `unique` برای اون انتخاب کنم این خطا بر طرف میشه :

```
6
7     @app.route('/home')
8     @app.route('/behnam')
9     def index():
10         name = "behnam"
11         return f"hello {name}"
12
13     @app.route('/behnam2')
14     def index2():
15         return "hello from behnam2"
```

```
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> flask run
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



در ارتباط با اسم فایلی که کد ها رو در اون می نویسید دقت کنید برای اینکه کامند
flask run کار کنه باید اسم فایل به صورت app.py یا ایتکه wsgi.py باشه به غیر از این دو اگر
اسمی باشه باید در محیط مشخص کنید که اسم اپلیکیشن چی هست

در لینوکس: FLASK_APP=behnam.py flask run

در ویندوز:

ابتدا باید متغیر FLASK_APP رو تعریف کنید بعد از طریق flask run command اپ رو
اجرا کنید

Set FLASK_APP=behnam.py

Flask run

در powershell :

```
$env:FLASK_APP = "behnam"
```

حالتی در flask وجود داره به معنای development که debugger را فعال کرده یعنی هر خطای
در اپ رخ دهد را نشان می دهد و هر موقع تغییری در اپ داشته باشیم نیازی که سرور را ریستارت کرده
خود به خود ریستارت شده و آماده به کار است

در لینوکس : FLASK_ENV=development flask run

در ویندوز :

Set FLASK_ENV=development

Flask run

در powershell :

```
$env:FLASK_ENV = "development"
```

Flask run

```
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic>
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> $env:FLASK_ENV="development"
(behnam) PS C:\Users\b.hasanbeygi\Desktop\flask\learn-flask\0-2 flask-basic> flask run
* Environment: development
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 889-583-553
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

اگر مشاهده کنید debug mode on و debugger اکتیو و هر موقع خطای در کد داشته باشید به شما اون خطا رو نشون میده.

خسته نباشید جلسه ی دوم هم تمام شد.

تاریخ ۴ ابان سال ۱۳۹۹