

Heads or tails!

نویسنده: بهنام حسن بیگی

MLops fanap soft /AIFA team

جلسه ی 5

فیدبک:یکی گفته بود چرا از gitignore استفاده نکردی تو پروژه هات که راست میگفت باید این کار رو انجام میدادم و انجام شد همون شخص هم گفته بود چرا اسم virtualenv تو گذاشتی behnam که یک بد آموزی هست برای دولوپر های جوون و نظم بسیار مهم:بعله درست میگی باید همیشه در کارت نظم وجود داشته باشه ولی من میخوام نشون بدم که میشه هر اسمی رو انتخاب کرد برای ساختن virtualenv و خودت رو نباید در قید اسم نگه داری با هر چیزی که راحت تری کارت رو پیش ببر.موفق باشید

تو جلسه ی قبل راجب templates صحبت کردیم یه مثال هم براش اجرا کردیم تو این جلسه میخوایم راجب template برای یک پروژه بنویسم اسم پروژه شیر یا خط heads or tails!!!
خب بریم سر اصل مطلب اول همون مراحل ثابت جلسات قبل رو دنبال کنید

۱ یک دایرکتوری برای پروژه بسازید 0-5 heads or tails

۲ داخل دایرکتوری یک virtual env با python3 تشکیل بده و فعالش کن

۳ library فلسک رو نصب کن

۴ یک نمونه کد بنویس و چک کن همه چیز اوکی هست یا نه!

خب بعد از انجام شدن مراحل بالا پروژه تو در حالت development اجرا کن و دو تا view بنویس یکی برای

Home paget و دیگری برای نتیجه شیر یا خطی که میخوای بنویسی results بدین صورت:

```

1  from flask import Flask, render_template
2  import random
3
4
5  app = Flask(__name__)
6
7
8  @app.route('/')
9  def home():
10     return render_template('home_page.html')
11
12  @app.route('/result/')
13  def heads_tails():
14     return render_template('headortails.html')

```

به library == random فعلا کاری نداشته باش برات توضیحش میدم.خب الان یک دایرکتوری تو دایرکتوری پروژه ت بساز اسمش رو بزار templates که محلی باشه واسه نگه داری پیج های html ات.بدین صورت:

OPEN EDITORS

0-5 TEMPLATES PART2

> __pycache__

> .vscode

> behnam

▼ templates

<> home_page.html

<> results.html

◆ .gitignore

📄 ~\$5 templates part2- heads or tails.docx

📄 0-5 templates part2- heads or tails.docx

📄 app.py

🖼 temp.PNG

🖼 view.PNG

🖼 همد یا تیلز.PNG

app.py

1 from flask import Flask, render_template

2 import random

3

4

5 app = Flask(__name__)

6

7

8 @app.route('/')

9 def home():

10 | return render_template('home_page.html')

11

12 @app.route('/result/')

13 def heads_tails():

14 | heads_tails = random.randint(1,2)

15 | return render_template('results.html')

دو تا فایل html در templates ایجاد کنید یکی homepage و دیگری results به این صورت باشند: (اسم headortails که برای html نوشته بودم رو به results تغییر دادم تا نتیجه ی بهتری بگیریم)

نگران حجم سنگین html نشید اگه extention برای html رو نصب کنید (تو قسمت extensions اگه سرچ کنی 5 html میبینی) با نوشتن کلمه html خودش 5 html رو نشون میده میتونی انتخابش کنی و این html کد بهت میده چیز زیاد پیچیده ای نیست حالا اگه بعدا وقت داشتیم frontend بهتون یاد میدم.

رای home page هم یک html کد بنویسید به این شکل:

```
templates > <> home_page.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
6      <title>home page</title>
7  </head>
8  <body>
9      <h1>
10         wellcome to head or tails web app
11     </h1>
12
13 </body>
14 </html>
```

خب الان باید شما دو تا فایل html برای دو تا view که ساختید داشته باشید و بتونید این دو تا رو ببینید.

قدم بعدی: برای نوشتن برنامه ی شیر یا خط باید library == random رو import کنیم این لایبرری به ما کمک میکنه که

اعداد random یا تصادفی برای ما ایجاد بشند به چه صورت؟! بدین شکل که بین یک بازه اعداد برای شما عدد تصادفی میسازد بدین صورت در results view کد برای ساختن app بنویسید:

```
1 from flask import Flask, render_template
2 import random
3
4
5 app = Flask(__name__)
6
7
8 @app.route('/')
9 def home():
10     return render_template('home_page.html')
11
12 @app.route('/result/')
13 def heads_tails():
14     heads_tails = random.randint(1,2)
15     return render_template('results.html')
```

پ.ن: تو ادامه ماجرا که کد رو نوشتم این اسم results برای خوشگلی تبدیل کردم به result الان حوصله نداشتم عکس بگیرم بازم و این اسم رو بنویسم result پس دچار پانیک نشید براتون هم فرقی نکنه که اسم نقی میشه یا تقی!!! کد رو اجرا کنید!!!

چرا بین دو عدد ۱ و ۲؟! چرا این بازه رو انتخاب کردیم؟! چون دو تا انتخاب داریم یا شیر میشه یا خط!!! پس یا عدد ۱ رو میبینی یا عدد ۲ رو.

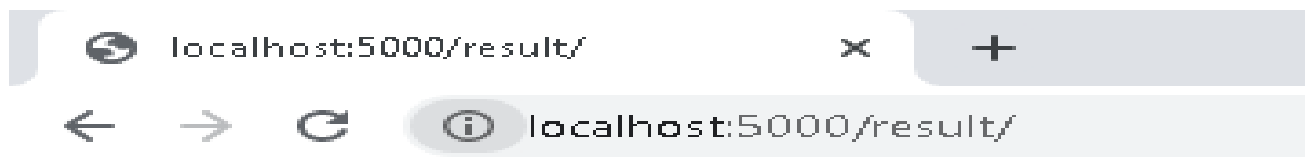
حالا بیایم این عدد رندوم رو ببینیم برای همین تو کد یکم تغییرات میدیم:

```

4
5  app = Flask(__name__)
6
7
8  @app.route('/')
9  def home():
10     return render_template('home_page.html')
11
12  @app.route('/result/')
13  def heads_tails():
14     heads_tails = random.randint(1,2)
15     return str(heads_tails)

```

به return دقت کنید که بعدش str رو نوشتیم باعث میشه که در view مرورگر ما اعداد نشون داده بشن

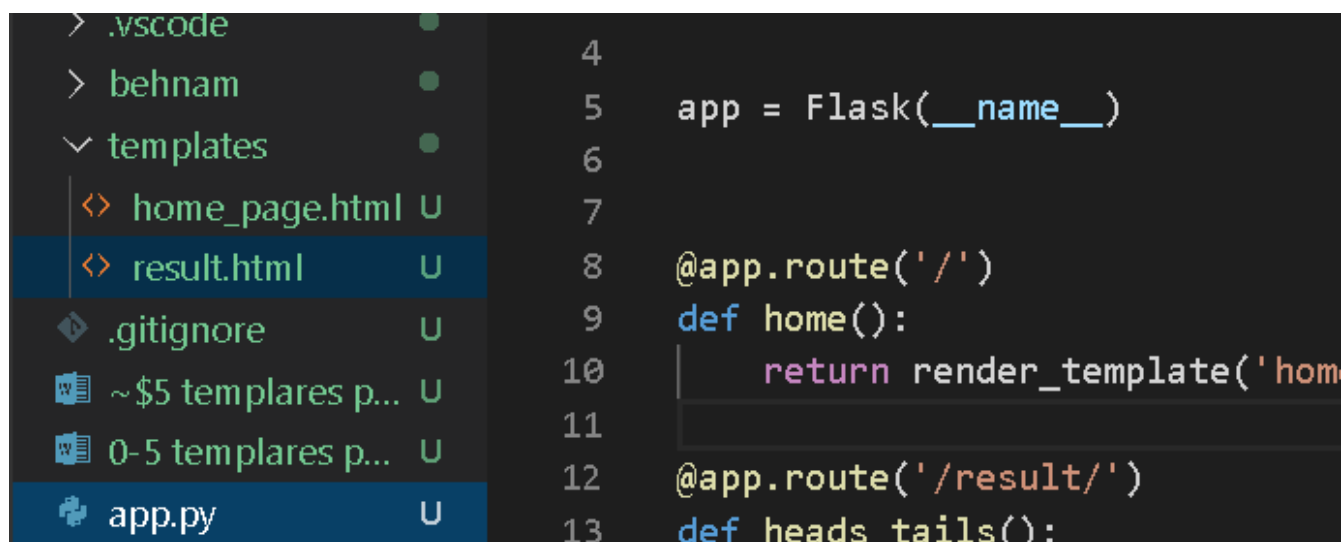


2

اگر صفحه رو رفرش کنید میتونید ببینید که این عدد هی تغییر میکنه!

الان ما نیاز داریم که یک template برای view == result که بتونیم از طریق html این متغییر و تغییرات رو ببینم پس برای همین میاییم از تابع render_template استفاده میکنیم و در templates یک html برای view == result

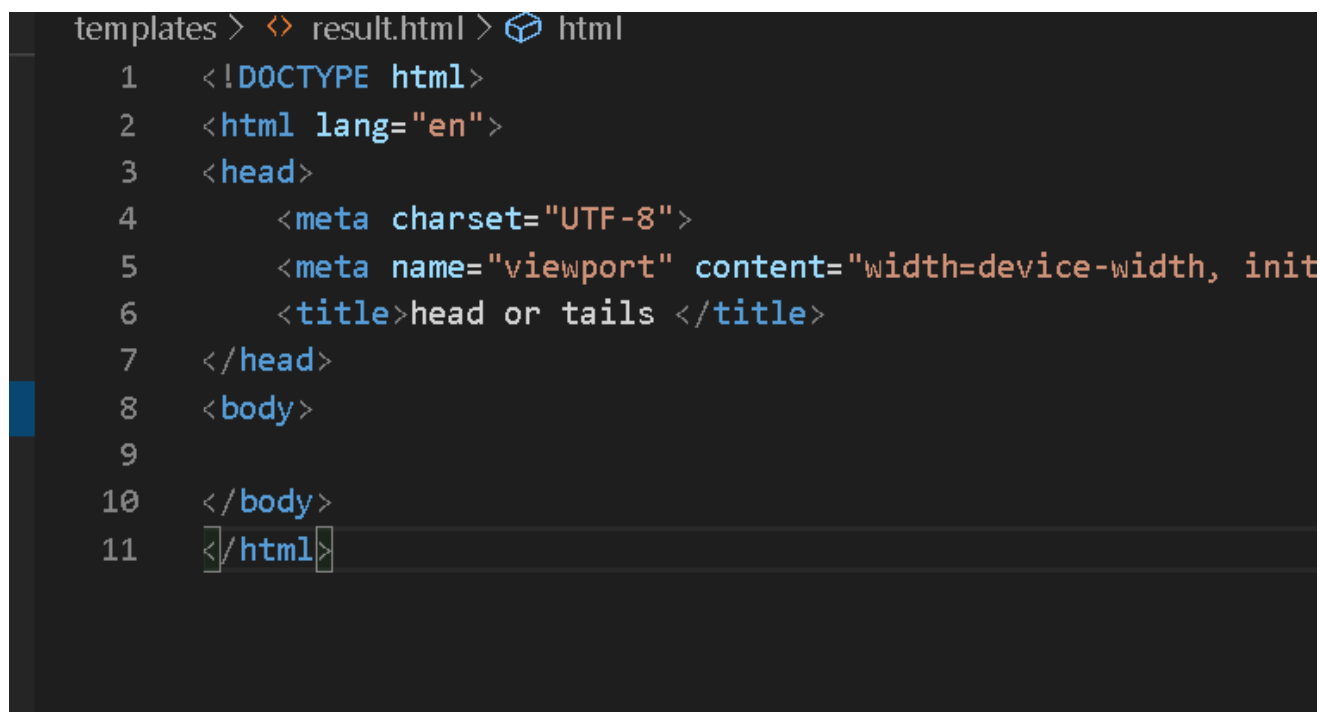
میسازیم .



```
> .vscode
> behnam
v templates
  <> home_page.html U
  <> result.html U
  .gitignore U
  ~$5 templates p... U
  0-5 templates p... U
  app.py U

4
5  app = Flask(__name__)
6
7
8  @app.route('/')
9  def home():
10     return render_template('home.html')
11
12  @app.route('/result/')
13  def heads tails():
```

فایل رو که تشکیل دادیم حالا میایم یک html کد برای این view می نویسیم که باعث بشه نتیجه کارمون رو بتونیم تو این view ببینم برای همین این html کد رو تو فایل result می نویسم به این شکل:



```
templates > <> result.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>head or tails </title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

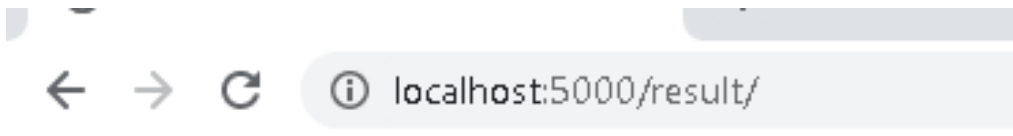
چون در کل من ادم خسته ی هستم extention برای html رو نصب کردم زمانی که میخوام html کد بنویسم به html مینویسم تو فایل بعد که html:5 رو شناختن Enter میکنم که به صورت دیفالت ساخته بشه برای همین اگه دقت کنید یک کد html طویل در اختیار ما قرار میده خب یک متن تو body می نویسم که بتونم ببینم ایا این کد من کار میکنه یا نه!!!

```
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>head or tails </title>
7 </head>
8 <body>
9     <h1> heads or tails</h1>
10 </body>
11 </html>
```

حالا میام این template که ساختم به فانکشن view م اضافه میکنم تا بتونم صفحه رو ببینم:

```
6
7
8 @app.route('/')
9 def home():
10     return render_template('home_page.html')
11
12 @app.route('/result/')
13 def heads_tails():
14     heads_tails = random.randint(1,2)
15     return render_template('result.html')
```

دقت کرده باشید با `render_template` فایل `html` رو اضافه کردم به `view` و بعد از سیو کردن کد میتونم برم ببینم که `view` من چطوری شده!



heads or tails

خب الان تو این مرحله باید کاری که کسی اومد تو صفحه `homepage` رو نتیجه یا `result` کلیک کرد بتونه بیاد به صفحه `result` و نتیجه رو ببینه برای همین به صفحه `homepage == html` میریم تا تغییرات رو اعمال کنیم:

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>home page</title>
7 </head>
8 <body>
9   <h1>
10    wellcome to head or tails web app
11  </h1>
12  <a href="{{ url_for('heads_tails') }}"> result</a>
13 </body>
14 </html>
```

ما در `html` یک `tag` داریم به اسم `a` که باعث میشه کلید یا یک `button` ایجاد بشه که ما رو به صفحه ی `html` کد که برای

Result نوشته انتقال بده یک نوع redirection ! در قسمت href ما از jinja2 == syntax استفاده کرده ایم که این syntax رو بهش میگن moustache syntax که moustache به معنی سیبیل هست اگر به کرلی براکت ها دقت کنید شبیه به سیبیل هستند!!! خب چرا این کار رو انجام میدیم چرا مستقیما نماییم اسم همون html کد که برای result رو نوشتیم قرار نمیدیم به این دلیل که اگر ما route مون تغییر پیدا کرد بتونه فایل html رو پیدا کنه و از اون استفاده کنه به طور مثال ما result رو اگر بکنیم results در route دیگه نمیتونه فایل html رو پیدا کنه و خطا میده برای همین میایم اسم فانکشن درون route رو میدیم که اسم فانکشن تغییر نمیدیم و همین باعث میشه اگر فردا /result شد results / کد ما توی دیوار نره!!!

خب الان ما از homepage خودمون یک مسیر رو باز کردیم به این شکل به صفحه ی html == result مون بدین شکل



بعد از کلیک روی result میره به صفحه ی result ما.

خب حالا ما میخوایم عدد ۱ و ۲ که همون heads or tails ما بودن رو در صفحه ی html == result ببینیم باید یک سری تغییرات بدیم.

```

templates > <> result.html > html > body > h1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial
6      <title>head or tails </title>
7  </head>
8  <body>
9      <h1> heads or tails</h1>
10     <h1> results : {{ heads_or_tails }}</h1>
11 </body>
12 </html>

```

باز هم از moustache سینتکس استفاده کردم ولی خب این دفعه اومدم یک متغیر رو به این html کد پاس دادم اگر فایل سیو کنید و برید تو مرورگر /result رو ببینید متوجه می شید که هیچ عددی رو جلوی results که نوشتید بهتون نشون نمیده چرا؟! افرین چون که هیچ مسیری رو مشخص نکرده اید که متغیری رو که تو backend هارد کد کردید بهتون نشون بده برای همین میرید تو کد backend یه تغییراتی رو اعمال میکنید!

```

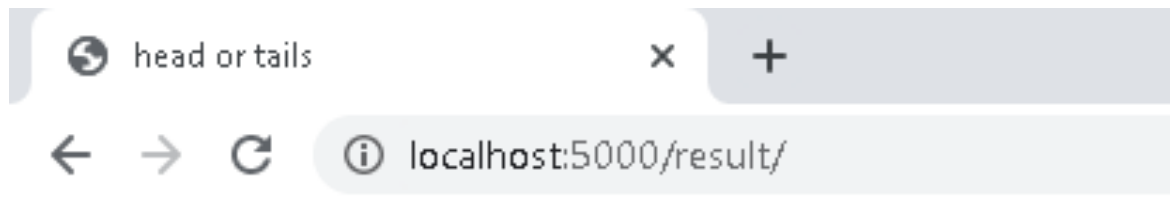
app = Flask(__name__)

@app.route('/')
def home():
    return render_template('home_page.html')

@app.route('/result/')
def heads_tails():
    heads_tails = random.randint(1,2)
    return render_template('result.html', heads_or_tails=heads_tails)

```

اگر دقت کرده باید یک پارامتر جلوی `render_template` پاس دادم که این همون اسمی که تو `html` برای `result` نوشته بودم توی `moustache` یعنی اینکه برای پاس دادن یک پارامتر یا متغیر یا نشون دادنش باید اسمی که اینجا پاس داده میشه با اسمی که به `html` کد داده باید یکی باشه و جلو این متغیر `heads_or_tails` اومدم متغیر اعدادمون قرار دادم اگر سیو کنم میتونم نتیجه رو ببینم.



heads or tails

results : 2

بدین شکل میشه یک متغیری رو که در `backend` هارد کردید رو به `frontend` پاس بدید. خب برای اون کسی که داره این بازی رو انجام میده معنی نمیده که بیاد ۲ رو ببینه یا ۱ رو اون فقط میاد کلمه ی `heads` یا `tails` رو میبینه برای همین باید کاری کنیم که این کلمات رو توی `html` کد ما دیده بشه پس:

با کمک کردن از `{% %}` می تونیم شرط خودمون رو توی `html` کد بنویسم یعنی اینکه بگیم خب اگر `heads_or_tails` رو که مشخص کردیم عدد ۱ داده بنویس `heads` اگر نه که بنویس `tails` در اخر هم حتما باید `endif` این شرط رو ببندیم وگرنه باز خطا میگیره میره تو دیوار !!!

```

    <meta name="viewport" content="width=device-width">
    <title>head or tails </title>
</head>
<body>
    <h1> heads or tails</h1>
    <h1> results :
        {% if heads_or_tails == 1%}
            heads
        {% else %}
            tails
        {% endif %}
    </h1>
</body>
</html>

```

خب الان اگر صفحه html رو ببینیم میتونیم بازی رو کامل انجام بدیم و نتیجه رو ببینیم.



heads or tails

results : heads

امید وارم خوشتون اومده باشه منو با فیدبک هاتون خوشحال کنید.

۱۸ ابان ۹۸