



# Python Practice

Добавление отображения курсов валют



# Python Practice

## Автор курса



Крементарь Ксения

Ведущий Python разработчик

Системный архитектор

в компании K-Solutions

# Python Practice

После урока обязательно



Повторите этот урок в видео формате на  
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на  
[TestProvider.com](http://testprovider.com)

## Добавление отображения курсов валют

# Python Practice

## На этом уроке

1. Спроектируем web сервис Golden Eye.
2. Познакомимся с основными элементами Flask — маршрутизация, шаблонизатор, конфигурирование сервиса, запуск приложения и т.п.
3. Добавим html страницу с отображением курсов валют, проверим ее работу в браузере.
4. Добавим обработку ошибок.

# Python Practice

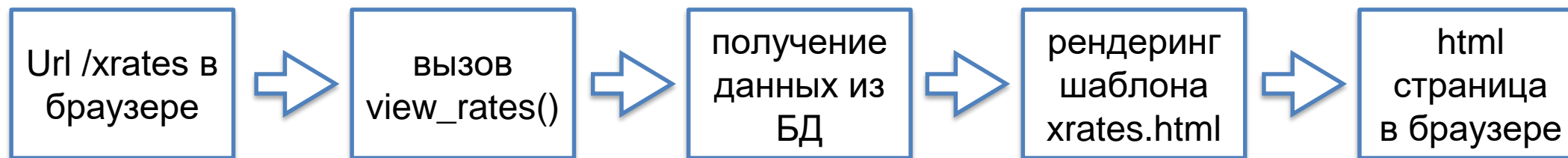
## Структура web проекта

- app.py — модуль, в котором осуществляется создание Flask приложения, создание экземпляра класса Flask.
- runserver.py — модуль, в котором происходит запуск приложения на тестовом локальном сервере.
- views.py — модуль, в котором описаны функции и соответствующие им url'ы приложения.
- controllers — модуль или пакет с функциями или классами, в которых описана основная логика работы.
- models — модуль или пакет с моделями для доступа к базе данных.
- config.py — модуль с настройками приложения.
- utils.py — модуль с утилитными функциями, которые используются из различных модулей.
- tests — модуль или пакет с тестами.
- Папка templates для хранения шаблонов приложения.

# Python Practice

## Логика работы отображения курсов

Выберем в качестве url — /xrates, view-функция — view\_rates



# Python Practice

## Шаблонизаторы

**Шаблонизатор** — это специальная библиотека, задача которой — генерация полноценного html документа, доступного для отображения в браузере. Генерация html документа осуществляется на основании шаблона и динамически передаваемых данных.

Для генерации html документа во Flask используется функция `render_template`. В качестве аргументов передается название шаблона и именованные аргументы, которые могут быть использованы в теле шаблона.

**Jinja2** — самый популярный шаблонизатор в языке программирования Python.



# Python Practice

## Основные элементы jinja2

`{{ ... }}` — для встраивания в шаблон любого Python выражения (например, переменной).

Конструкция **for**

`<ul>`

`{% for user in users %}`

`<li>{{ user }}</li>`

`{% endfor %}`

`</ul>`

Конструкция **if**

`{% if users %}`

`<ul>`

`{% for user in users %}`

`<li>{{ user }}</li>`

`{% endif %}`

`</ul>`

`{% endif %}`

# Python Practice

## Шаблон xrates.html

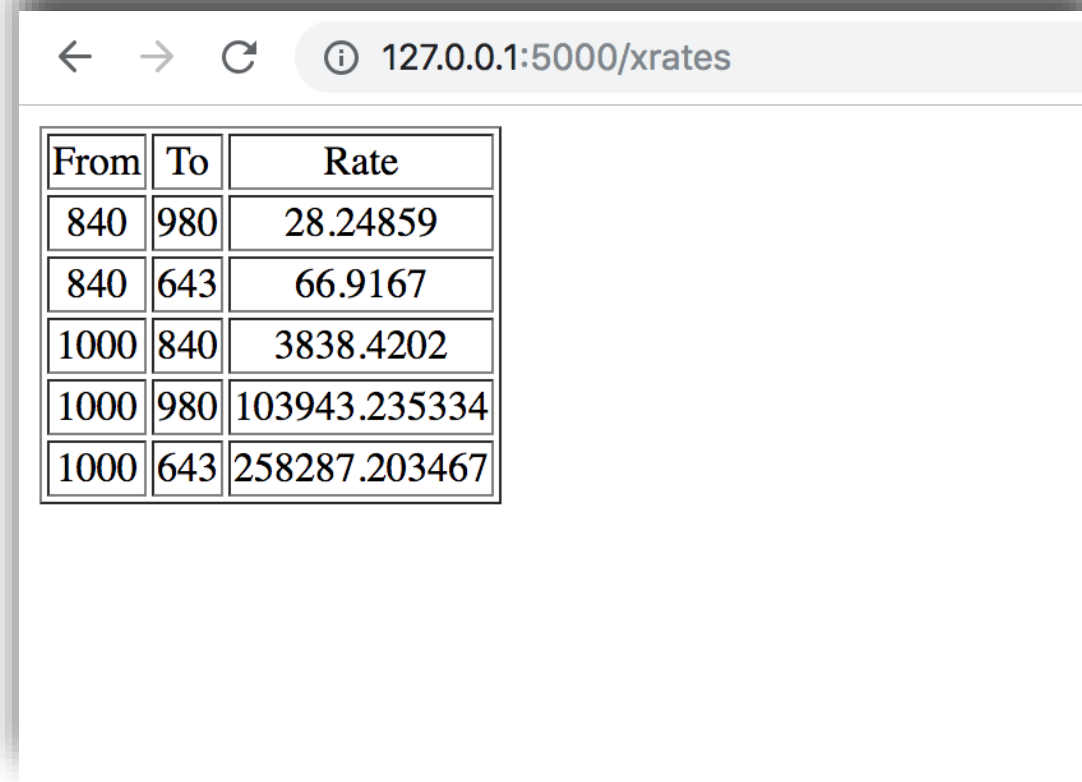
```
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5     <title>Rates</title>
6 </head>
7 <body>
8     {% if not xrates %}
9     <h1>No rates!</h1>
10    {% endif %}
11
12    <table border="1">
13        <thead align="center">
14            <td>From</td><td>To</td><td>Rate</td>
15        </thead>
16        {% for rate in xrates %}
17        <tr align="center">
18            <td>{{rate.from_currency}}</td>
19            <td>{{rate.to_currency}}</td>
20            <td>{{rate.rate}}</td>
21        </tr>
22        {% endfor %}
23    </table>
24 </body>
25 </html>
26
```

```
1 from flask import render_template
2
3 from models import XRate
4
5
6 def get_all_rates():
7     xrates = XRate.select()
8     return render_template("xrates.html", xrates=xrates)
9
```

# Python Practice

## Шаблон xrates.html

```
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Rates</title>
6 </head>
7 <body>
8   {% if not xrates %}
9   <h1>No rates!</h1>
10  {% endif %}
11
12  <table border="1">
13    <thead align="center">
14      <td>From</td><td>To</td><td>Rate</td>
15    </thead>
16    {% for rate in xrates %}
17    <tr align="center">
18      <td>{{rate.from_currency}}</td>
19      <td>{{rate.to_currency}}</td>
20      <td>{{rate.rate}}</td>
21    </tr>
22    {% endfor %}
23  </table>
24 </body>
25 </html>
26
```



From	To	Rate
840	980	28.24859
840	643	66.9167
1000	840	3838.4202
1000	980	103943.235334
1000	643	258287.203467

# Python Practice

## Обработка ошибок

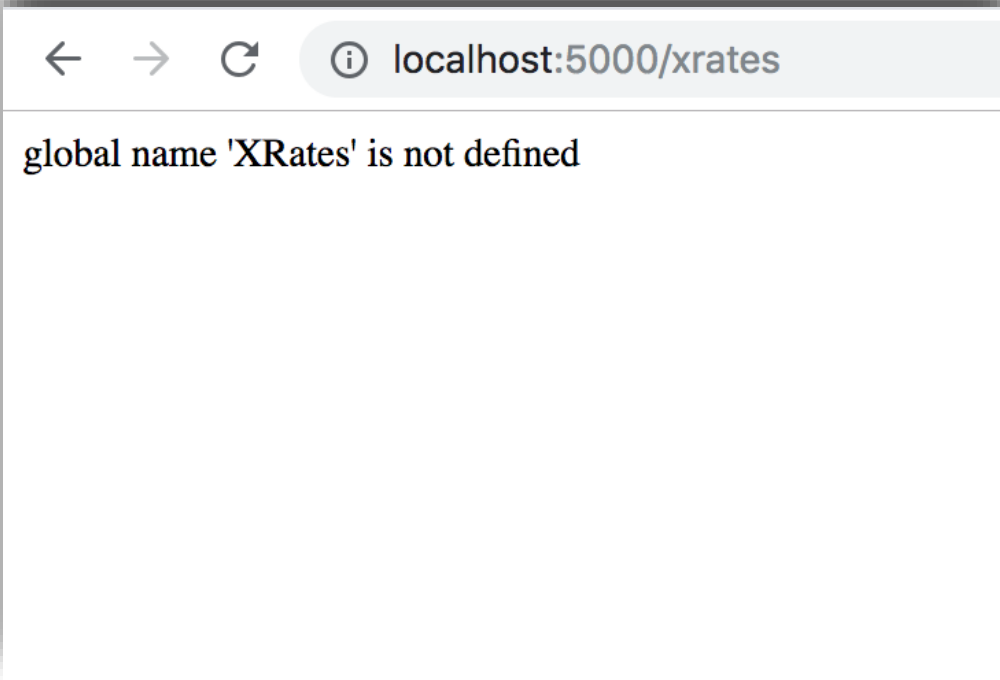
### NameError

NameError: global name 'XRates' is not defined

#### Traceback (most recent call last)

```
File "/Library/Python/2.7/site-packages/flask/app.py", line 2000, in __call__
    return self.wsgi_app(environ, start_response)
File "/Library/Python/2.7/site-packages/flask/app.py", line 1991, in wsgi_app
    response = self.make_response(self.handle_exception(e))
File "/Library/Python/2.7/site-packages/flask/app.py", line 1567, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/Library/Python/2.7/site-packages/flask/app.py", line 1988, in wsgi_app
    response = self.full_dispatch_request()
File "/Library/Python/2.7/site-packages/flask/app.py", line 1641, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/Library/Python/2.7/site-packages/flask/app.py", line 1544, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/Library/Python/2.7/site-packages/flask/app.py", line 1639, in full_dispatch_request
    rv = self.dispatch_request()
File "/Library/Python/2.7/site-packages/flask/app.py", line 1625, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/Users/krementar/projects/itvdn_course/lesson 9/views.py", line 14, in view_rates
    return controllers.get_all_rates()
File "/Users/krementar/projects/itvdn_course/lesson 9/controllers.py", line 8, in get_all_rates
    xrates = XRates.select()
```

NameError: global name 'XRates' is not defined



# Python Practice

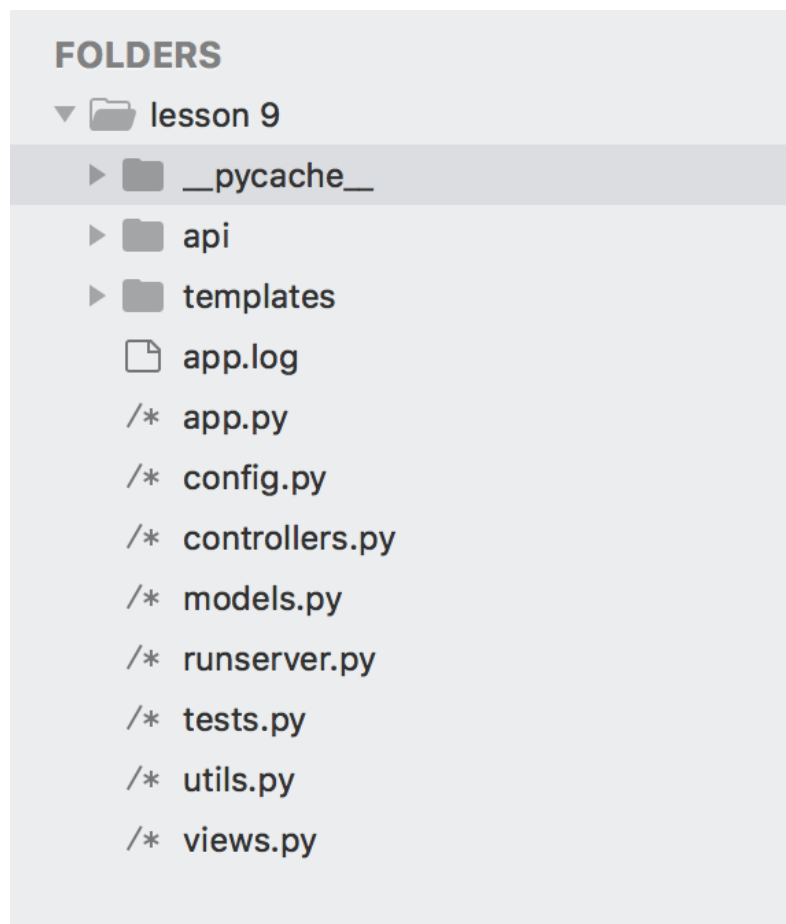
## Обработка ошибок

По умолчанию, если обработка запроса во Flask приложении завершилась без ошибок, клиенту возвращается HTTP статус 200, который свидетельствует о корректной обработке запроса.

Flask позволяет явно указать HTTP код обработки запроса, метод `make_response` принимает в качестве аргументов (body, status, headers), где body — тело ответа, строка, status — HTTP статус ответа и headers — словарь заголовков ответа. С помощью функции `make_response` можно реализовать обработку ошибки, возникшей при обработке запроса. Принято возвращать код HTTP код 500, в случае если работа приложения завершилась с ошибкой.

# Python Practice

## Структура проекта Golden Eye



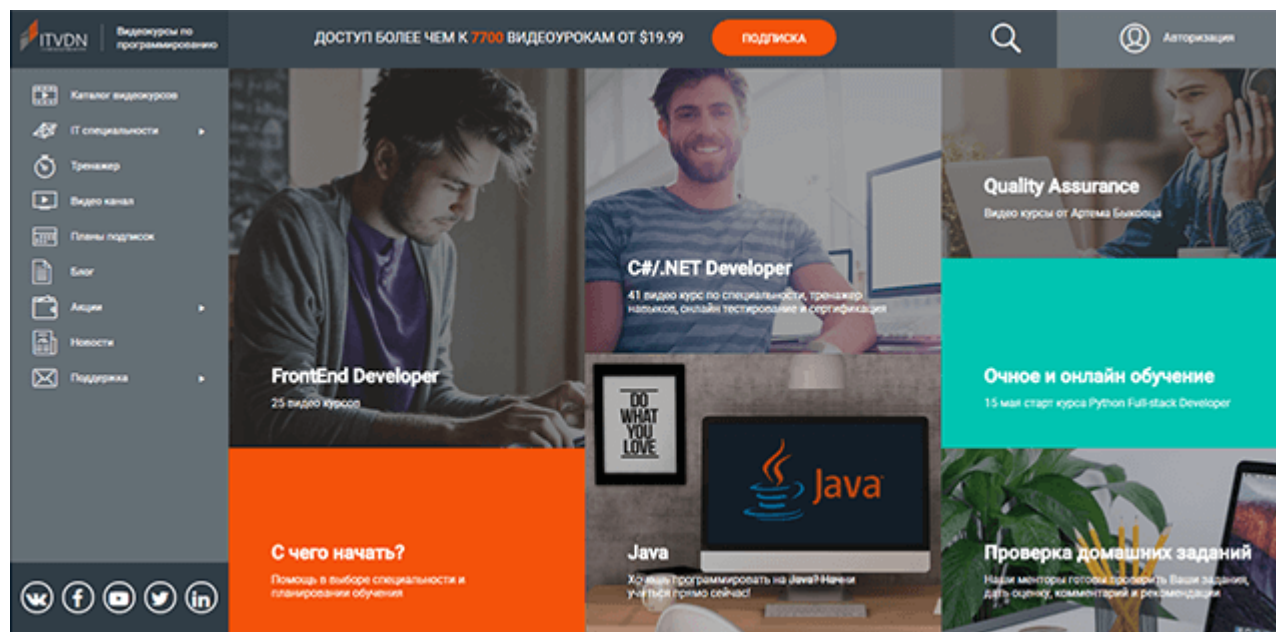
# Python Practice

## Что дальше?

Добавление методов для получения данных о курсах по api. То есть разработка своего собственного полноценного api для удаленных клиентов, которые смогут вызывать его для получения информации о курсах валют.

# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://ITVDN.com) для закрепления пройденного материала.

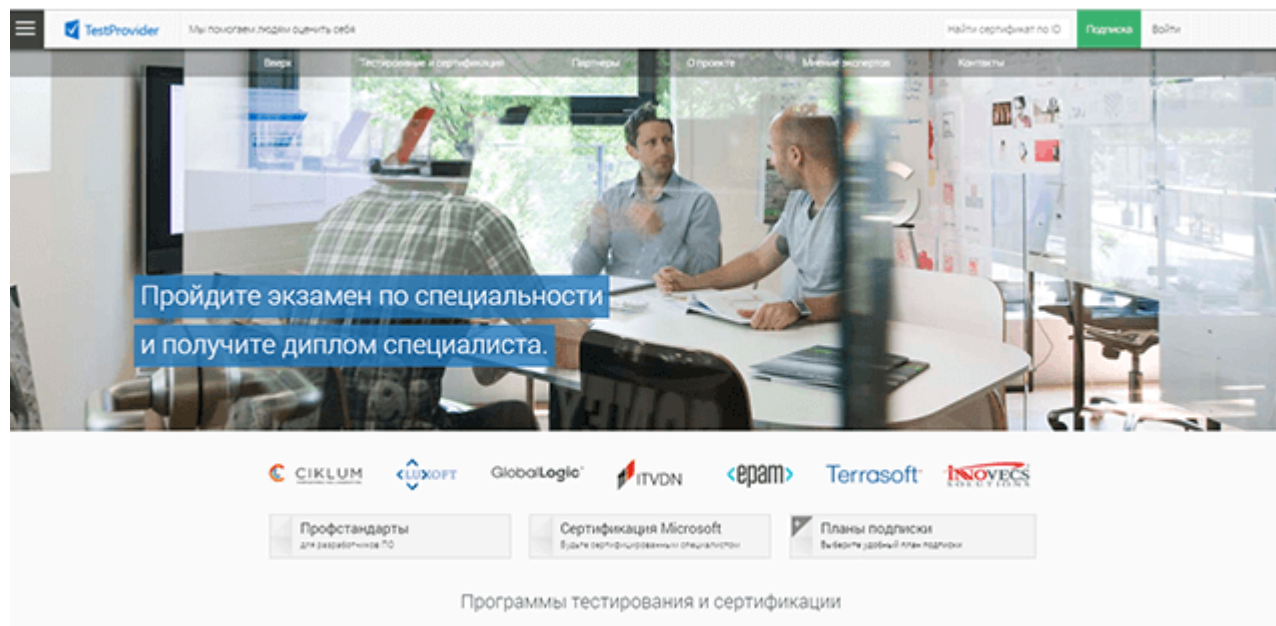
Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.





# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# Python Practice

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

