Практика: простая обработка запросов

Задания

1. По корневому url’у (**/**). Отобразите следующую информацию:

Текст:

|  |
| --- |
| <h1>"Изучаем django"</h1>  <strong>Автор</strong>: <i>**Иванов И.П**.</i> |

Вместо **Иванов И.И.** - ваши данные.

***Совет***: вспомните про f-строки.

1. По адресу: **/about** отображаем такой результат в браузере:

|  |
| --- |
| Имя: **Иван**  Отчество: **Петрович**  Фамилия: **Иванов**  телефон: **8-923-600-01-02**  email: **vasya@mail.ru** |

Данные о пользователе храним в python-коде(в переменных). Так, чтобы при изменении данных информация обновилась по всем url-адресам.

***Примечание***: в данном задании достаточно вернуть минимальный HTML для отображения ответа в указанном виде.

1. Создаем список произвольных товаров:

|  |
| --- |
| items = [  {"id": 1, "name": "Кроссовки abibas" ,"quantity":5},  {"id": 2, "name": "Куртка кожаная" ,"quantity":2},  {"id": 5, "name": "Coca-cola 1 литр" ,"quantity":12},  {"id": 7, "name": "Картофель фри" ,"quantity":0},  {"id": 8, "name": "Кепка" ,"quantity":124},  ] |

***Примечание***: **items** - глобальная переменная в файле **views.py**

По url **/item/1** - отобразить на странице информацию о товара с id=1 (название, кол-во)  
По url **/item/2** - отобразить на странице информацию о товара с id=2  
и т.д.

***Примечание***: на каждой странице необходимо сделать минимальное HTML-оформление.  
***Подсказка***: чтобы не плодить множество однотипных функций, посмотрите [тут](https://django.fun/docs/django/ru/3.0/topics/http/urls/#example).

1. По url **/item/10** отображаем “**Товар с id=10 не найден**”, если товара с таким id не существует.
2. По url **/items** отображаем нумерованный список всех товаров.
3. Добавляем к списку товаров гиперссылки, ведущие на страницу соответствующего товара.   
   ***Совет***: как оформить гиперссылку [смотри тут](http://htmlbook.ru/html/a).  
   На каждой странице делаем ссылку “назад к списку товаров”.