Работа с контекстом

Context в шаблонах

```
context = {
    'user': request.user,
    'products': [
        {'name': 'Худи черного цвета', 'price': 1950},
        {'name': 'Широкие штаны', 'price': 2300},
        {'name': 'Обувь Nike', 'price': 4900},
    ]
}
```

Тип данных контекста - словарь (dict)

Обычно если мы хотим передать некие данные в шаблон (страницу), то данные сохраняются в некий объект и потом этот объект “достается” в шаблоне и отображается на странице.
В рамках джанго этот объект представляет из себя словарь.
Например, можно создать ключ юзер и добавить ему какое-то значение, а дальше можно обращаться уже ключу и значение отобразиться непосредственно в шаблене.

Откройте файл views.py
В контроллере индекс создадим тестовый контекст

```
def index(request):
    context = {'title': 'Test Title',
               'username': 'sveta',
               }
    return render(request, "products/index.html", context)
```

Переменные (placeholders)

Предположим, у нас есть какой-то контекст в котором лежат следующие данные

```
context = {'first_name': 'John', 'last_name': 'Doe'}
```

Если мы хотим получить оттуда данные, то мы просто вставляем ключ и получаем значение
Код:

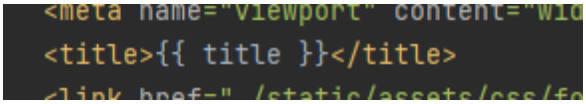
```
My first name is {{ first_name }}. My last name is {{ last_nam
e }}.
```

Вывод:

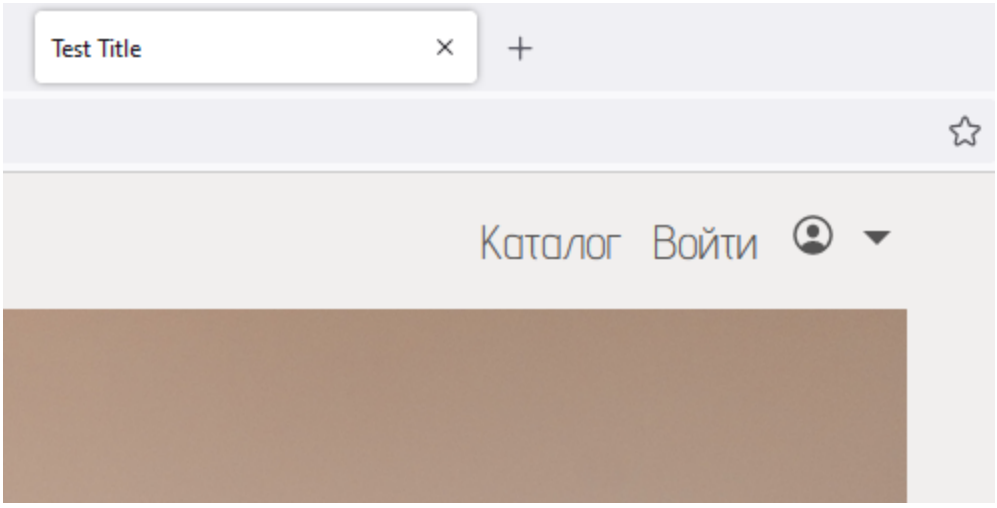
```
My first name is John. My last name is Doe.
```

p.s. две фигурные скобки это синтаксис джанго

Откройте index.html
Добавим туда title из контекста



Название вкладки должно смениться



Смените Test Title на ваше название :)

Теги

Если бы условие писали просто в питоне, то можно было использовать стандартную конструкцию. Теперь как это работает в джанго.
Например, есть контекст с ключом is_promotion (есть ли сейчас акция на товар) со значением по умолчанию true

```
context = {'is_promotion': True}
```

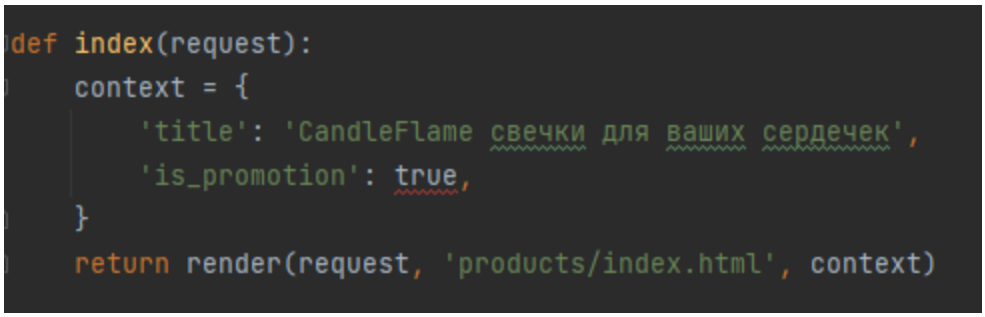
В html-коде прописываем, что если акция активна, то выводим определенные значения.

```
{% if is_promotion %} Promotion is turned on. {% endif %}
```

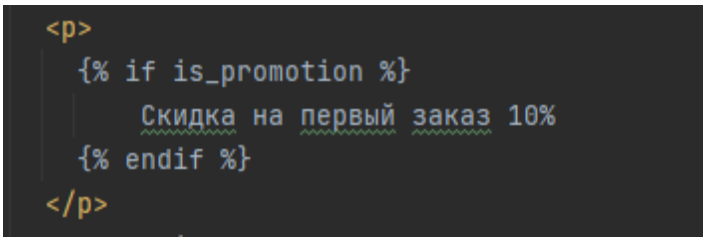
Вывод:

```
Promotion is turned on.
```

Откройте views.py



Откройте index.html
Сообщение про скидку обернем в этот тег (если у вас нет скидки, оберните любую фразу)



Если вы откроете сайт - ничего не изменится.

Если значение поменять на False

```python
def index(request):
    context = {
        'title': 'CandleFlame свечки для ваших сердечек',
        'is_promotion': False,
    }
    return render(request, 'products/index.html', context)
```

то ваше сообщение обернутое в этот тег пропадет.

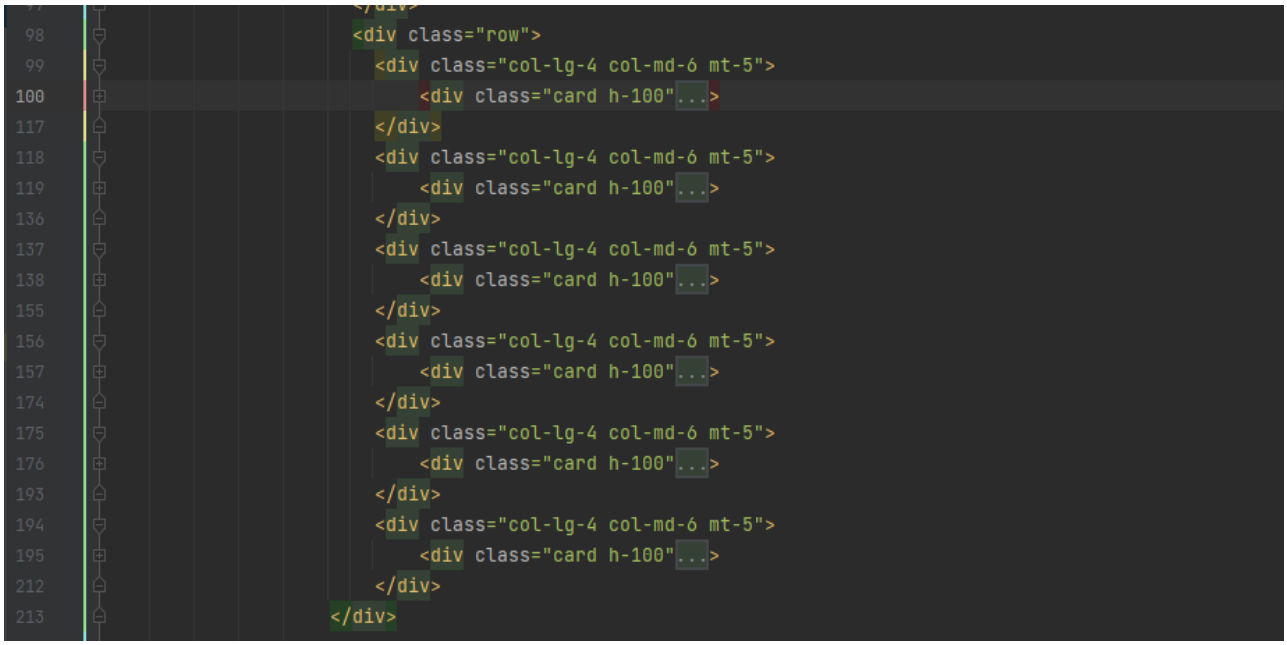
Все, все лишнее можно стирать, оставьте только title/

```python
def index(request):
    context = {
        'title': 'CandleFlame свечки для ваших сердечек',
    }
    return render(request, 'products/index.html', context)
```

Откройте views.py, нам нужен контроллер products.
Создадим в нем контекст и добавим title

```python
def products(request):
    context = {
        'title': 'CandleFlame - Каталог',
    }
    return render(request, 'products/products.html', context)
```

Далее откройте products.html (ваша страница с каталогом) и обратите внимание, что здесь несколько одинаковых элементов (карточки товаров) и логично, что мы не должны выводить каждую карточку товара по-отдельности.



p.s. строки свернуты

В идеале должен быть цикл, который выводит все данные.
Внутри контекста создайте ключ products. Внутри будет список со словарями:

```
def products(request):
    context = {
        'title': 'CandleFlame - Каталог',
        'products': [
            {}
        ]
    }
    return render(request, 'products/products.html', context)
```

Какие данные там будут:

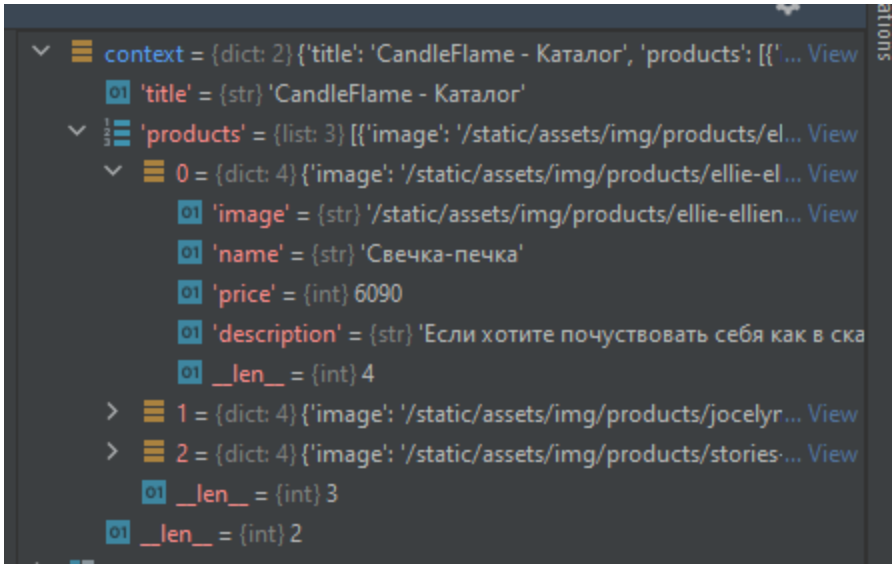
- Путь до фото
- Название
- Цена
- Описание

```
<div class="col-lg-4 col-md-6 mt-5">
    <div class="card h-100">
        <a href="#">
            <img class="card-img-top"
                 src="/static/assets/img/products/ellie-ellien-HWqsW7FLWf4-unsplash.jpg"
                 alt="">
        </a>
        <div class="card-body">
            <h4 class="card-title">
                <a href="#">Свечка-печка</a>
            </h4>
            <h5>6 090,00 руб.</h5>
            <p class="card-text">Если хотите почуствовать себя как в сказке.</p>
        </div>
        <div class="card-footer text-center">
            <button type="button" class="btn btn-outline-dark">Отправить в корзину</button>
        </div>
    </div>
</div>
```

Добавьте три словаря (мы делаем это для общего понимания, поэтому больше не надо. Дальше это будет выводиться через БД)

```
def products(request):
    context = {
        'title': 'CandleFlame - Каталог',
        'products': [
            {
                'image': '/static/assets/img/products/ellie-ellien-HWqsW7FLWf4-unsplash.jpg',
                'name': 'Свечка-печка',
                'price': 6090,
                'description': 'Если хотите почуствовать себя как в сказке.'
            },
            {
                'image': '/static/assets/img/products/jocelyn-morales-KmL31BZp9qQ-unsplash.jpg',
                'name': 'Свечка-хлебопечка',
                'price': 6090,
                'description': 'Дома как в булошной.'
            },
            {
                'image': '/static/assets/img/products/stories-mFTOFzUKdl0-unsplash.jpg',
                'name': 'Свечка-сердечка',
                'price': 6090,
                'description': 'Для романтического настроения.'
            },
        ]
    }
    return render(request, 'products/products.html', context)
```

Если написать код контекста в консоль (это вы можете не делать), то в pycharm можно увидеть структуру словаря:



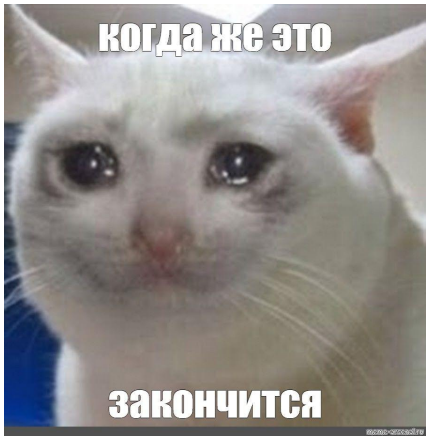
Пример запроса по ключевому слову

```
>>> context['title']
'CandleFlame - Каталог'
```

Если, например, хотим вывести все имена

```
>>> for product in context['products']:
...     print(product['name'])
...
Свечка-печка
Свечка-хлебопечка
Свечка-сердечка
```

Но естественно в шаблонах это работает не так :)



Никогда.

Открываем products.html

1. Передаем туда title
2. Карточки с товарами сверните

```
<div class="row">
    <div class="col-lg-4 col-md-6 mt-5"...>
    <div class="col-lg-4 col-md-6 mt-5"...>
    <div class="col-lg-4 col-md-6 mt-5"...>
    <div class="col-lg-4 col-md-6 mt-5"...>
    <div class="col-lg-4 col-md-6 mt-5"...>
    <div class="col-lg-4 col-md-6 mt-5"...>
</div>
```

3. Закомментируйте все, кроме первой (чтобы пока не удалять)
4. Первую карточку товара оберните в тег

```
<div class="row">
    {% for product in products %}
        <div class="col-lg-4 col-md-6 mt-5"...>
    {% endfor %}
```

5. Раскройте ее и теперь надо заменить значения на переменные:

```
{% for product in products %}
    <div class="col-lg-4 col-md-6 mt-5">
      <div class="card h-100">
          <a href="#">
              <img class="card-img-top"
                   src="{{ product.image }}"
                   alt="">
          </a>
          <div class="card-body">
              <h4 class="card-title">
                  <a href="#">{{ product.name }}</a>
              </h4>
              <h5>{{ product.price }}</h5>
              <p class="card-text">{{ product.description }}</p>
          </div>
          <div class="card-footer text-center">
              <button type="button" class="btn btn-outline-dark">Отправить в корзину</button>
          </div>
      </div>
    </div>
{% endfor %}
```

Все, запустите/обновите страницу. У вас должны были сформироваться три карточки товара.