# Лабораторная работа: Реализация передачи данных из представлений в шаблоны

## Цель работы:

Научиться передавать данные из представлений в шаблоны в Django для динамического отображения информации на веб-страницах.

## Описание задания:

Создать Django-приложение с несколькими представлениями, передавать различные типы данных (переменные, списки, объекты модели) из представлений в шаблоны и отобразить их на страницах.

## Инструменты:

- Установленная среда Python
- Установленный Django
- Текстовый редактор (например, PyCharm, Visual Studio Code)

## Ход работы:

### 1. Создание проекта и приложения Django:

Создайте новый проект Django, используя команду:
```bash
django-admin startproject myproject
cd myproject
```
Создайте новое приложение, например `myapp`, командой:
```bash
python manage.py startapp myapp
```
Добавьте `myapp` в список `INSTALLED\_APPS` в `settings.py`:
```python
# myproject/settings.py
INSTALLED\_APPS = [
 ...,
 'myapp',
]
```

### 2. Создание модели для передачи данных:

В приложении `myapp` откройте файл `models.py` и создайте модель `Product`, которая будет содержать информацию о товарах, такие как `name`, `price`, и `description`:
```python
# myapp/models.py
from django.db import models

class Product(models.Model):
 name = models.CharField(max\_length=100)
 price = models.DecimalField(max\_digits=10, decimal\_places=2)
 description = models.TextField()

 def \_\_str\_\_(self):
 return self.name
```
Выполните миграции для создания таблицы `Product` в базе данных:
```bash
python manage.py makemigrations
python manage.py migrate
```

### 3. Добавление данных в админ-панель:

Зарегистрируйте модель `Product` в `admin.py` для управления записями через интерфейс администратора:
```python
# myapp/admin.py
from django.contrib import admin
from .models import Product

admin.site.register(Product)
```
Создайте суперпользователя для доступа в админ-панель:
```bash
python manage.py createsuperuser
```
Запустите сервер, войдите в админ-панель и добавьте несколько записей для модели `Product`.

### 4. Создание представления с передачей данных в шаблон:

В `views.py` создайте представление `product\_list`, которое передает список всех товаров из базы данных в шаблон:
```python
# myapp/views.py
from django.shortcuts import render
from .models import Product

def product\_list(request):
 products = Product.objects.all()
 return render(request, 'myapp/product\_list.html', {'products': products})
```

### 5. Создание шаблона для отображения данных:

Создайте директорию `templates/myapp` и добавьте туда файл `product\_list.html`.
Внутри `product\_list.html` отобразите информацию о каждом товаре, используя переданные данные:
```html
<!-- templates/myapp/product\_list.html -->
<h1>Список товаров</h1>
<ul>
 {% for product in products %}
 <li>
 <h3>{{ product.name }}</h3>
 <p>Цена: {{ product.price }} USD</p>
 <p>{{ product.description }}</p>
 </li>
 {% endfor %}
</ul>
```

### 6. Передача и отображение простых переменных в шаблоне:

В `views.py` создайте представление `simple\_view`, которое передает простую переменную, например, текущее время, в шаблон:
```python
# myapp/views.py
from datetime import datetime

def simple\_view(request):
 current\_time = datetime.now()
 return render(request, 'myapp/simple\_view.html', {'current\_time': current\_time})
```
Создайте шаблон `simple\_view.html` для отображения переданной переменной:
```html
<!-- templates/myapp/simple\_view.html -->
<h1>Текущее время</h1>
<p>{{ current\_time }}</p>
```

### 7. Добавление маршрутов URL для представлений:

В `urls.py` создайте маршруты для `product\_list` и `simple\_view`, чтобы можно было получить к ним доступ через URL-адреса:
```python
# myapp/urls.py
from django.urls import path
from . import views

urlpatterns = [
 path('products/', views.product\_list, name='product\_list'),
 path('time/', views.simple\_view, name='simple\_view'),
]

# myproject/urls.py
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
 path('admin/', admin.site.urls),
 path('', include('myapp.urls')),
]
```

### 8. Проверка работы:

Запустите сервер и откройте страницы `/products/` и `/time/` в браузере, чтобы убедиться, что данные корректно передаются из представлений в шаблоны и отображаются на странице.

## Контрольные вопросы:

- Как передать данные из представления в шаблон в Django?

- Что такое контекст шаблона и как он используется для передачи данных?

- Какие типы данных можно передавать из представления в шаблон?

- Как отобразить элементы списка данных, переданных из представления в шаблон?

## Дополнительное задание:

Создайте новое представление и шаблон для отображения подробной информации о товаре. Используйте параметр `pk` в маршруте URL для передачи идентификатора товара в представление и выводите информацию о конкретном товаре.