**1. Как передать данные в шаблон?**

В Django: данные передаются в шаблон через контекст в функции render.

В Flask: данные передаются через аргумент context в функции render\_template.

Пример для Django:

from django.shortcuts import render

def my\_view(request):

context = {'name': 'Alina', 'age': 22}

return render(request, 'my\_template.html', context)

Пример для Flask:

from flask import Flask, render\_template

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

context = {'name': 'Alina', 'age': 22}

return render\_template('my\_template.html', \*\*context)

**2. Какие данные можно передать при рендеринге шаблона?**

При рендеринге шаблона можно передать любые данные: строки, числа, списки, словари, объекты и т.д. Эти данные будут доступны в шаблоне.

**3. Как обозначается переменная в шаблоне? Как она связана с переменными в Python?**

В шаблоне переменные обозначаются с использованием двойных фигурных скобок {{ variable\_name }}. Эти переменные связаны с переменными в контексте, переданном из Python.

<p>Name: {{ name }}</p>

<p>Age: {{ age }}</p>

**4. Покажите конструкцию блока c условием в шаблоне.**

{% if age >= 18 %}

<p>{{ name }} is an adult.</p>

{% else %}

<p>{{ name }} is not an adult.</p>

{% endif %}

**5. Покажите конструкцию блока с циклом шаблоне.**

<ul>

{% for student in students %}

<li>{{ student.name }} - {{ student.age }}</li>

{% endfor %}

</ul>

**6. Как создать свой фильтр для данных в шаблоне**

 В Django: для создания пользовательского фильтра нужно создать файл templatetags и зарегистрировать фильтр.

 В Flask (Jinja2): нужно зарегистрировать фильтр через app.template\_filter.

**Пример для Django:**

# Создайте файл myapp/templatetags/custom\_filters.py

from django import template

register = template.Library()

@register.filter(name='uppercase')

def uppercase(value):

return value.upper()

**В шаблоне:**

{% load custom\_filters %}

<p>{{ name|uppercase }}</p>

**Пример для Flask:**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.template\_filter('uppercase')

def uppercase\_filter(value):

return value.upper()

@app.route('/')

def index():

context = {'name': 'Alina'}

return render\_template('my\_template.html', \*\*context)

**В шаблоне:**

<p>{{ name|uppercase }}</p>