

ШАБЛОНЫ И ФОРМЫ В FLASK

Спикер

Немков Максим Юрьевич

СОДЕРЖАНИЕ ТЕМЫ

→ Введение в Jinja2. Основные элементы шаблонов Jinja2

→ Операторы и выражения

→ Работа с формами

→ Наследование и включение шаблонов

JINJA2

это важный инструмент для создания динамических веб-страниц с Python. Он позволяет вставлять переменные, условия, циклы и другие элементы прямо в HTML код

РАБОТА С ПЕРЕМЕННЫМИ

Основным элементом в Jinja2 является переменная. Для того чтобы вывести переменную на страницу, используются **двойные фигурные скобки**

```
<body>
  <h1>Привет, {{ name }}!</h1>
</body>
```

Определение понятия

ВЫРАЖЕНИЯ

это конструкции, которые состоят
из операторов и операндов

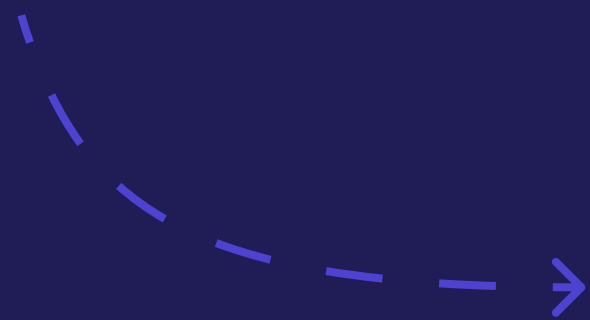
Определение понятия

ОПЕРАТОРЫ

это символы или слова, которые выполняют определённые действия над операндами

РАБОТА С ПЕРЕМЕННЫМИ

Давайте рассмотрим
несколько примеров
выражений с разными
операторами



```
...  
{{ 5 * 3 }}
```

Умножение

```
...  
{{ 6 > 4 }}
```

Сравнение

```
...  
{% set x = 5 %}
```

Присваивание

УСЛОВНЫЙ ОПЕРАТОР

Условный оператор в Jinja2 позволяет выполнять определённые действия в зависимости от значения переменной или выражения

Это может быть полезно, когда вы хотите отобразить различный контент в зависимости от условий

```
{% if condition1 %}  
    <div>Первое условие истинно</div>  
{% elif condition2 %}  
    <div>Второе условие истинно</div>  
{% else %}  
    <div>Ни одно из условий не истинно</div>  
{% endif %}
```

Здесь ``condition`` — это выражение, которое оценивается как истинное или ложное. Если оно истинно, будет выполнен блок кода внутри ``{% if %}``

ЦИКЛ FOR

Конструкция `for` позволяет перебирать элементы списка или другого итерируемого объекта

```
...  
{% for item in list %}  
    {{ item }}  
{% endfor %}
```

Здесь `list` — это список элементов, которые будут выведены на экран. Конструкция `for/endifor` будет повторять блок кода для каждого элемента списка

Определение понятия

ФОРМЫ

это способ взаимодействия пользователя
с сайтом или приложением

ФОРМЫ

Здесь мы создали форму с полем ввода имени и кнопкой отправки

→ Тег ``action`` указывает на URL-адрес, куда будут отправлены данные формы после ее отправки

→ А метод ``method`` определяет, каким образом будут отправлены данные — через GET или POST

```
<form action="/submit" method="POST">
  <label for="name">Имя:</label>
  <input type="text" id="name" name="name"><br>
  <button type="submit">Отправить</button>
</form>
```

Атрибут ``name``,
чтобы указать имя
поля при отправке
данных на сервер

НАСЛЕДОВАНИЕ ШАБЛОНОВ

- ✓ Наследование шаблонов позволяет нам создать базовый шаблон, который содержит общую структуру и логику, а затем наследовать от него другие шаблоны, добавляя или изменяя нужный нам контент
- ✓ Включение шаблонов позволяет нам включать один шаблон в другой

НАСЛЕДОВАНИЕ ШАБЛОНОВ

Пример наследования шаблонов в Jinja2

```
# Базовый шаблон base.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Базовый шаблон</title>
  </head>
  <body>
    {% block content %}{% endblock %}
  </body>
</html>
```


НАСЛЕДОВАНИЕ ШАБЛОНОВ

Теперь давайте создадим наследуемый шаблон

```
{% extends "base.html" %}

{% block content %}
    <h1>Страница</h1>
    {{ message }}
{% endblock %}
```

Здесь мы используем тег ``extends`` для наследования от базового шаблона ``base.html``. Мы также заполняем блок ``content`` своим собственным содержимым, включая заголовок и сообщение

ПОДВЕДЕМ ИТОГИ

- ✓ Узнали про шаблоны и формы
- ✓ Поработали с операторами и выражениями
- ✓ Изучили что такое формы
- ✓ Узнали про наследование шаблонов

