SDEV 2401

Rapid Backend App Development

Url Templates and Views - 1

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments

- No Cheating

- Be a good classmate

- Don’t waste your time

- Show up to class

WE ARE ESSENTIAL TO ALBERTA | NAIT

Agenda

On the right is what we will cover today.

- Jinja2 Template Fundamentals
- Rendering Variables in Templates
- Loops and Conditionals
- Filters and Comments
- For Loop Special Variables
- Debugging Templates
- Summary

WE ARE ESSENTIAL TO ALBERTA | NAIT

Jinja2 Template Fundamentals

- Jinja2 is a powerful templating engine used in Python web frameworks like Flask, Django, and FastAPI.

- Templates allow you to dynamically generate content by combining static markup with variables and logic.

- Common use cases: web pages, emails, configuration files, and more.

WE ARE ESSENTIAL TO ALBERTA | NAIT

Rendering Variables in Templates

- Variables are rendered using double curly braces: `{{ variable }}`
- Example:

```
from django.template import Template, Context

data = {"animal": "cat", "sound": "meow"}
template = Template("A {{ animal }} says {{ sound }}.")
context = Context(data)
print(template.render(context))
```

- Output: `A cat says meow.`

WE ARE ESSENTIAL TO ALBERTA | NAIT

Loops and Conditionals

- Use `{% for %}` and `{% if %}` for logic in templates.
- Example:

```
data = {
    "fruits": [
        {"name": "Apple", "color": "red"},
        {"name": "Banana", "color": "yellow"},
        {"name": "Grape", "color": "purple"},
    ]
}
template = Template("""
{% for fruit in fruits %}
  {% if fruit.color == "yellow" %}
    {{ fruit.name }} is yellow!
  {% else %}
    {{ fruit.name }} is not yellow.
  {% endif %}
{% endfor %}
""")
```

- Output:

```
Apple is not yellow.
Banana is yellow!
```

Filters and Comments

- Filters modify data: `{{ name | upper }}`
- Example:

```
template = Template("{{ city | lower }}")
context = Context({"city": "LONDON"})
print(template.render(context))  # Output: london
```

- Comments in templates: `{# This is a comment #}`

WE ARE ESSENTIAL TO ALBERTA | NAIT

For Loop Special Variables

- `forloop.counter` : 1-based index
- `forloop.first` : True if first item
- Example:

```
template = Template("""
{% for n in numbers %}
  {% if forloop.first %}First: {% endif %}
  {{ forloop.counter }}. {{ n }}
{% endfor %}
""")
context = Context({"numbers": [10, 20, 30]})
```

- Output:

```
First: 1. 10
2. 20
3. 30
```

WE ARE ESSENTIAL TO ALBERTA | NAIT

Debugging Templates

- Use `{% debug %}` to print all context variables.
- Example:

```
template = Template("""
{% debug %}
""")
context = Context({"foo": "bar"})
print(template.render(context))
```

WE ARE ESSENTIAL TO ALBERTA | NAIT

Summary

- Jinja2 templates allow variable rendering, logic, filters, and comments.
- Practice with your own data and templates to get comfortable.
- Next: Integrate templates with Django views for dynamic web pages.

WE ARE ESSENTIAL TO ALBERTA | NAIT

Example

Let’s go run a few examples together