Solution: For Loop in Jinja

In this lesson, we will be taking a look at the solution of the challenge presented in the previous lesson.

WE'LL COVER THE FOLLOWING

The complete implementation of the problem is provided below. Let's take a look at it!

Solution

```
"""Flask Application for Paws Rescue Center."""
from flask import Flask, render_template
app = Flask(__name__)

"""Information regarding the Pets in the System."""
pets = [
            {"id": 1, "name": "Nelly", "age": "5 weeks", "bio": "I am a tiny kitten rescued b
            {"id": 2, "name": "Yuki", "age": "8 months", "bio": "I am a handsome gentle-cat.
            {"id": 3, "name": "Basker", "age": "1 year", "bio": "I love barking. But, I love
            {"id": 4, "name": "Mr. Furrkins", "age": "5 years", "bio": "Probably napping."},
        ]

@app.route("/")
def homepage():
    """View function for Home Page."""
    return render_template("home.html", pets = pets)


@app.route("/about")
def about():
    """View function for About Page."""
    return render_template("about.html")



if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=3000)
```

Explanation

Let’s take a look at how we solved this problem.

Modification in `app.py`

In `app.py`, at **line 16**, we returned the `pets` dictionary from the `homepage` view. This will enable us to access this variable in the `home.html` template.

Modification in `home.html`

The bulk of the solution is present in the `home.html` template.

1. First, we used the `for` loop syntax in jinja to ***traverse*** through the `pets` dictionary.
2. Then, we added a new *table row* in the loop that renders all columns of a single `pet`. We used the Python syntax between `{{ }}` to get the values from the dictionary.
3. Lastly, the most **tricky part** of this challenge was to get the `filename` for the image associated with each `pet`. For this purpose, we used the `pet["id"]` variable and converted it into a `string` using the `string` filter in Jinja. Then, this `id` was appended to the string `".jpg"` to create the complete `filename`.

In the next challenge, we will be creating a dynamic route to view the details of each pet individually.