

Variables

In this lesson, we will get to know the method of passing variables to the template and using them with Jinja!

WE'LL COVER THE FOLLOWING ^

- Introduction
- How does it work?
- `String` example
- `Dictionary` example

Introduction

Flask allows us to pass any Python object to the template, which can then be referred to inside the template using Jinja syntax.

How does it work?

We can use variables inside the templates by following these two steps:

1. The object is provided as a named argument to the `render_template()` function.

```
return render_template("index.html", my_object = Object)
```

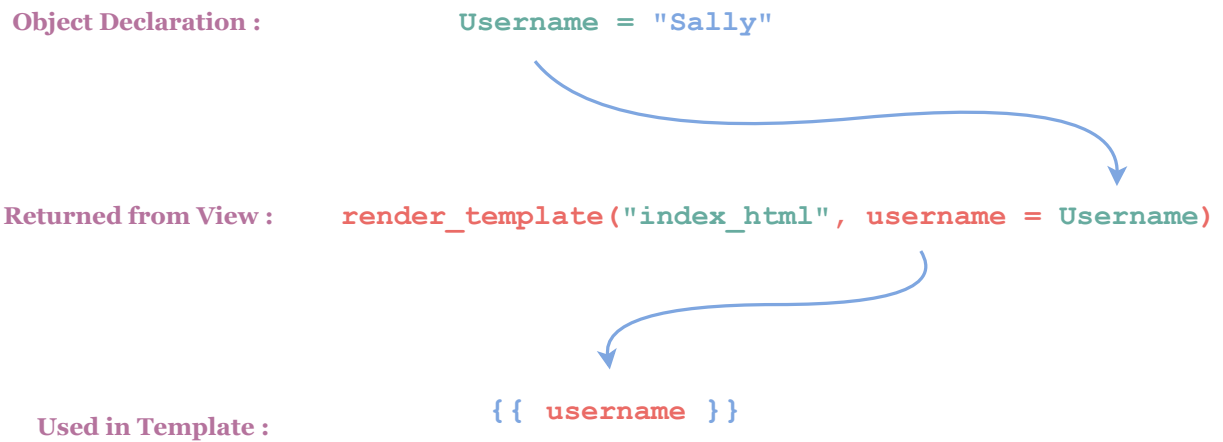
2. The value of this object is then fetched inside the template using `{{ }}` syntax.

```
{{ my_object }}
```

🔗 **Note:** Multiple parameters can be passed to the template using the method mentioned above.

String example

Consider a string variable called `Username`. We can access its value inside the template using the following steps:



Steps to use a String object inside the Template

This example is implemented below.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Jinja2 Demo</title>
</head>
<body>
  <h3>Username:</h3>
  <p>
    {{username}}
  </p>
</body>
</html>
```

Dictionary example

Consider a dictionary type variable called `Users`. We can access its value inside the template using the following steps:

Object Declaration:

```
Users = {  
    Archie: Amsterdam,  
    Veronica: London,  
    Betty: San Francisco,  
    Jughead: Los Angeles  
}
```

Returned from View:

```
render_template("index_html", users = Users)
```

Used in Template:

```
{{ users['Betty'] }}
```

Steps to use a Dictionary object inside the Template

This example is implemented below.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="utf-8">  
    <title>Jinja2 Demo</title>  
</head>  
<body>  
    <h3>Betty lives in:</h3>  
    <p>  
        {{ users["Betty"] }}  
    </p>  
</body>  
</html>
```

Quick Quiz!

Q

Which one of these statements will print the first index of a list called `arr`.

COMPLETED 0%



1 of 1



In the next lesson, we will look at the syntax for **loops** and **conditional** statements in Jinja!