PYTHON SNIPPETS

Nested String Interpolation in Python

Phil Best
3 min read · Sep 9

In last week's Python snippet we took a look at some formatting options for numbers using Pythons Format Specification Mini Language. This week we're going to continue with formatting, but this time we're going to look at nested string interpolation.

String interpolation basics

Just so we're all on the same page, string interpolation is where we put a string inside another string. For example, like this:

```
name = "Jose"
print(f"Hello, {name}")  # Hello, Jose
```

We can also use the `format` method:

```
name = "Jose"
print("Hello, {}".format(name))  # Hello, Jose
```

The result is exactly the same, f-strings are just a newer syntax available from Python 3.6 onward.

In both versions we take the string `"Hello, {}"` , where the curly braces are a placeholder. Then we interpolate (insert) the string assigned to the variable `name` , replacing the placeholder curly braces.

Nested string interpolation

An interesting thing is that we can actually perform string interpolation inside one of these placeholder curly braces. Now, why on Earth would you ever want to do that?

Well, for one, it allows us to dynamically set formatting options when using the Format Specification Mini Language. For example, perhaps we are dynamically creating file names with includes a numbering scheme, and we want our program to be as generic as possible. We therefore perhaps don't want to hardcode the numbering scheme.

We're going to use a simplified example here, where the other aspects of the filename never change, but you could easily replace those aspects as well.

First, the hardcoded version, so we know what to compare it to:

```python
number_of_files = 3

for file_number in range(1, number_of_files + 1):
    print(f"image{file_number:03}.png")

# image001.png
# image002.png
# image003.png
```

Now the nested version:

```
number_of_files = 3
number_digits = int(input("How many digits are used in the numbering

for file_number in range(1, number_of_files + 1):
    print(f"image{file_number:0{number_digits}}.png")
```

Now if the user enters 5, we'll get a naming scheme like this: `image00001.png`.

We could make various improvements to this, such as adding a `try` / `except` block for the user input to catch invalid values, but this is enough to demonstrate the concept.

Nested string interpolation with `format`

We can also perform the same kind of nested string interpolation with the `format` method.

If we use positional arguments, the nested placeholders are considered to come directly after the placeholder they inhabit. So the example from above would look like this using positional arguments:

```
number_of_files = 3
number_digits = int(input("How many digits are used in the numbering

for file_number in range(1, number_of_files + 1):
    print("image{:0{}}.png".format(file_number, number_digits))
```

I think this is a little confusing, so if you have to use `format` for some reason, I'd opt for keyword arguments like this:

```
number_of_files = 3
number_digits = int(input("How many digits are used in the numbering

for file_number in range(1, number_of_files + 1):
    print("image{number:0{padding_amount}}.png".format(
        number=file_number,
        padding_amount=number_digits
    ))
```

This way there can be no confusion about what belongs where. That being said, f-strings are clearly the much cleaner option in this case if you can use them.

Wrapping Up

I hope you enjoyed this Python snippet post and learnt something new. It's a bit of a niche topic this week, but it's something that a lot of people don't know about, and it's certainly very useful in some cases.

As always, if you're looking to upgrade your Python skills, we recommend you take a look at our Complete Python Course! It takes you from absolute beginner, right up to advanced Python concepts, so it's a great way to really dive into Python programming.

You might want to also sign up to our mailing list below, as we'll be posting coupons each month so that you can get the best deals on all our courses.

Phil Best

I'm a freelance developer, mostly working in web development. I also create course content for Teclado!

Read More

Python Snippets

Assignment expressions in Python →

Python's namedtuples →

Python's divmod Function →

→ See all 26 posts

GUI DEVELOPMENT

Creating Snake Using Tkinter's Canvas Widget (Part 2)

In this post we finish off our Snake game using Tkinter's Canvas widget. Learn how to move objects on the Canvas, delete them, or add more.

Phil Best
16 min read · Sep 12

Like what you see? Enter your e-mail to hear when new posts come out!

E-mail*

Name*

Receive our newsletter and get notified about new posts we publish on the site, as well as occasional special offers.

Sign Up