String formatting

September 22, 2022

1 String formatting

In many of the notebooks in this series of lessons, you'll see something like this:

```
name = 'Cody'
print(f'Hello, {name}!')
# Hello, Cody!
```

This is called "string formatting," and it's using a tool called "f-strings," which are available in Python 3.6 and later distributions. It's one way to pass variables to a template string. Note the `f` prepended to the string and the curly brackets `{}` placeholder with the name of the variable you'd like to inject into the string.

Here's another example:

[ ]:
```
my_name = 'Cody'
my_age = 37
my_state = 'South Dakota'
```

[ ]:
```
greeting = f'Hello, my name is {my_name}. I am {my_age} years old, and I live␣
  ↪in {my_state}.'
```

[ ]:
```
print(greeting)
```

Another way to do the same thing is to use the `.format()` string method, which is a little more verbose:

[ ]:
```
greeting_2 = 'Hello, my name is {}. I am {} years old, and I live in {}.'

print(greeting_2.format(my_name, my_age, my_state))
```

Using f-strings is cleaner, generally, but `format()` can be more versatile in some situations because you can create a template string *before* the variable exists.

Here's an example of what I mean. Using the `format()` method:

[ ]:
```
file_template = '{year}-data.csv'
```

[ ]:
```
for y in range(1990, 2000):
    print(file_template.format(year=y))
```

That approach wouldn't work with an f-sring:

```
[ ]: file_template = f'{year}-data.csv'
```

1.0.1 Formatting numbers

Just like in Excel, you can change the formatting of a piece of data for display purposes without changing the underlying data itself. Here are a couple of the more common recipes for formatting numbers:

```
[ ]: my_number = 1902323820.823
```

Add thousand-separator commas

```
[ ]: f'{my_number:,}'
```

Increase or decrease decimal precision

```
[ ]: # no decimal places
     f'{my_number:.0f}'
```

```
[ ]: # two decimal places
     f'{my_number:.2f}'
```

```
[ ]: # two decimal places ~and~ commas
     f'{my_number:,.2f}'
```

```
[ ]: # add a dollar sign to that - note that it's OUTSIDE of the curly brackets
     f'${my_number:,.2f}'
```

```
[ ]: # add a british pound sign to that
     f' {my_number:,.2f}'
```

```
[ ]: # add an emoji to that
     f' {my_number:,.2f}'
```

```
[ ]: # add an emoji to that ... in a sentence
     f'I have  {my_number:,.2f} in GrimaceCoin, my new cryptocurrency.'
```