Python Fstring Numeric Decimal and Significance Formatting

Fan Wang

2020-12-14

Contents

1 FString

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

```
import numpy as np
import string as string
import random as random
```

1.1 Use FString to get String of variable

1.2 Fstring Print Strings with Numeric Values and Other Strings

After some code segment, print some outputs declaring the end of operation and print results. See Getting the name of a variable as a string.

```
st_some_str_var = "abcefg"
st_some_str_var_name = f'{st_some_str_var=}'.split('=')[0]
print(f'{isinstance(st_some_str_var_name, str)=}')
```

```
## isinstance(st_some_str_var_name, str)=True
```

```
print(f'{st_some_str_var_name=}')
```

```
## st_some_str_var_name='st_some_str_var'
```

1.3 Fstring Format Floating Point Values Significance

Keep up to X significance numbers. See How can I print many significant figures in Python?.

```
# birthweight estimate
fl_estimate_birthweight = 3181.49
# 1 significance number
print(f'1 significance numbers, {fl_estimate_birthweight:.1}')
# 3 significance number
```

```
## 1 significance numbers, 3e+03
```

```python
print(f'3 significance numbers, {fl_estimate_birthweight:.3}')
# 5 significance number
```

```
## 3 significance numbers, 3.18e+03
```

```python
print(f'5 significance numbers, {fl_estimate_birthweight=:.5}')
# 7 significance number
```

```
## 5 significance numbers, fl_estimate_birthweight=3181.5
```

```python
print(f'7 significance numbers, {fl_estimate_birthweight=:.7}')
```

```
## 7 significance numbers, fl_estimate_birthweight=3181.49
```

1.4 Fstring Format Floating Point Values Decimals

Keep up to X decimal points.

```python
# birthweight estimate
fl_estimate_birthweight = 3181.49
# 1 significance number
print(f'0 decimal, {fl_estimate_birthweight:.0f}')
# 3 significance number
```

```
## 0 decimal, 3181
```

```python
print(f'1 decimal, {fl_estimate_birthweight:.1f}')
# 5 significance number
```

```
## 1 decimal, 3181.5
```

```python
print(f'2 decimals, {fl_estimate_birthweight=:.2f}')
# 7 significance number
```

```
## 2 decimals, fl_estimate_birthweight=3181.49
```

```python
print(f'3 decimals, {fl_estimate_birthweight=:.3f}')
```

```
## 3 decimals, fl_estimate_birthweight=3181.490
```

1.5 Fstring Decimal Function

For a regression table, use a combination of significance number and decimal number formatting to properly output information. For numbers that exceed

```python
# Define a formatter function
def fstring_formater(st_float, it_decimal):
    # strip the string float, and format with it_decimal number of decimals
    st_float = st_float.strip()
    fl_float = float(st_float)
    st_float_rounded = f'{fl_float:.{it_decimal}f}'
    return st_float_rounded

# Print
f'{fstring_formater("1.2345", 3)=}'
```

```
## 'fstring_formater("1.2345", 3)=\'1.234\''
```

```python
f'{fstring_formater("1.2345", 2)=}'
```

```
## 'fstring_formater("1.2345", 2)=\'1.23\''
```

```python
f'{fstring_formater(" 1.2345 ", 1)=}'
```

```
## 'fstring_formater(" 1.2345 ", 1)=\'1.2\''
```

```python
f'{fstring_formater(" 1.2345", 1)=}'

# Alternatively
```

```
## 'fstring_formater(" 1.2345", 1)=\'1.2\''
```

```python
it_decimal = 3
fl_float = 123.456789
print(f'Formating with .{it_decimal}f -> {fl_float:.{it_decimal}f}')
```

```
## Formating with .3f -> 123.457
```

1.6 Fstring Decimal Function with Dictionary Threshold

Specify a dictionary of threshold levels, where numbers below a particular threshold level, and above the previous one, will be formatted with a particular number of decimal points. This is to be used for example, in formatting a table, where numbers above say 1000 will have 0 decimal points, but numbers below 100 should have 1 decimal point, etc.

The example below also demonstrates how to break out of the inner loop to proceed with the outter loop. See Breaking out of nested loops.

In the example below, we loop through a list of numbers of different size with different decimals, and the *dc_round_decimal* dictionary determines how each number is formatted based on their size.

```python
ls_st_numformated = []
ls_fl_num2format = [0.0012345, 0.12345, 12.345, 123.45, 1234.5, 123456.789]
# dict incremental formatter
# if below 0.1 keep 4 decimals, If below 1 keep 3, if below 100 keep 2,
# if otherwise above, then keep 0 decimals
dc_round_decimal = {0.1:4, 1:3, 100:2, float("inf"):0}
# Loop over formatter
for fl_num2format in ls_fl_num2format:
    for fl_threshold, it_decimals in dc_round_decimal.items():
        if fl_num2format <= fl_threshold:
            st_float_rounded = f'{fl_num2format:.{it_decimals}f}'
            ls_st_numformated.append(st_float_rounded)
            print(f'{fl_num2format=}, {st_float_rounded=}, {fl_threshold=}, {it_decimals=}')
            break
    else:
        continue
# print return
```

```
## fl_num2format=0.0012345, st_float_rounded='0.0012', fl_threshold=0.1, it_decimals=4
## fl_num2format=0.12345, st_float_rounded='0.123', fl_threshold=1, it_decimals=3
## fl_num2format=12.345, st_float_rounded='12.35', fl_threshold=100, it_decimals=2
## fl_num2format=123.45, st_float_rounded='123', fl_threshold=inf, it_decimals=0
## fl_num2format=1234.5, st_float_rounded='1234', fl_threshold=inf, it_decimals=0
## fl_num2format=123456.789, st_float_rounded='123457', fl_threshold=inf, it_decimals=0
```

```
print(f'{ls_st_numformated=}')
```

```
## ls_st_numformated=['0.0012', '0.123', '12.35', '123', '1234', '123457']
```