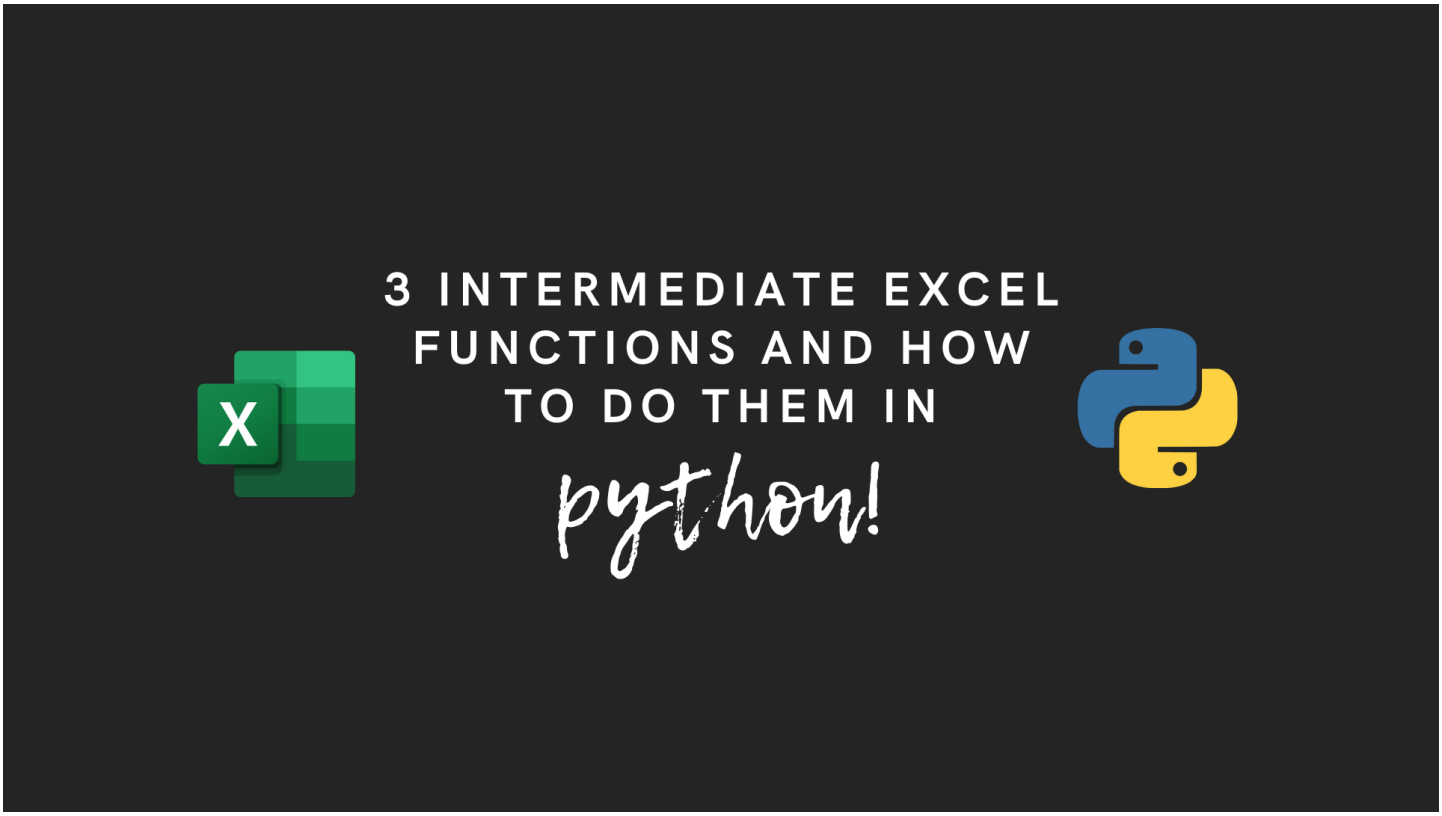
# 3 Intermediate Excel Functions and How to do Them in Python!

Do more than test the waters in Pandas!

Nik Piepenbreier
Jun 8 · 3 min read ★




Let's get going with Pandas! Source: Nik Piepenbreier

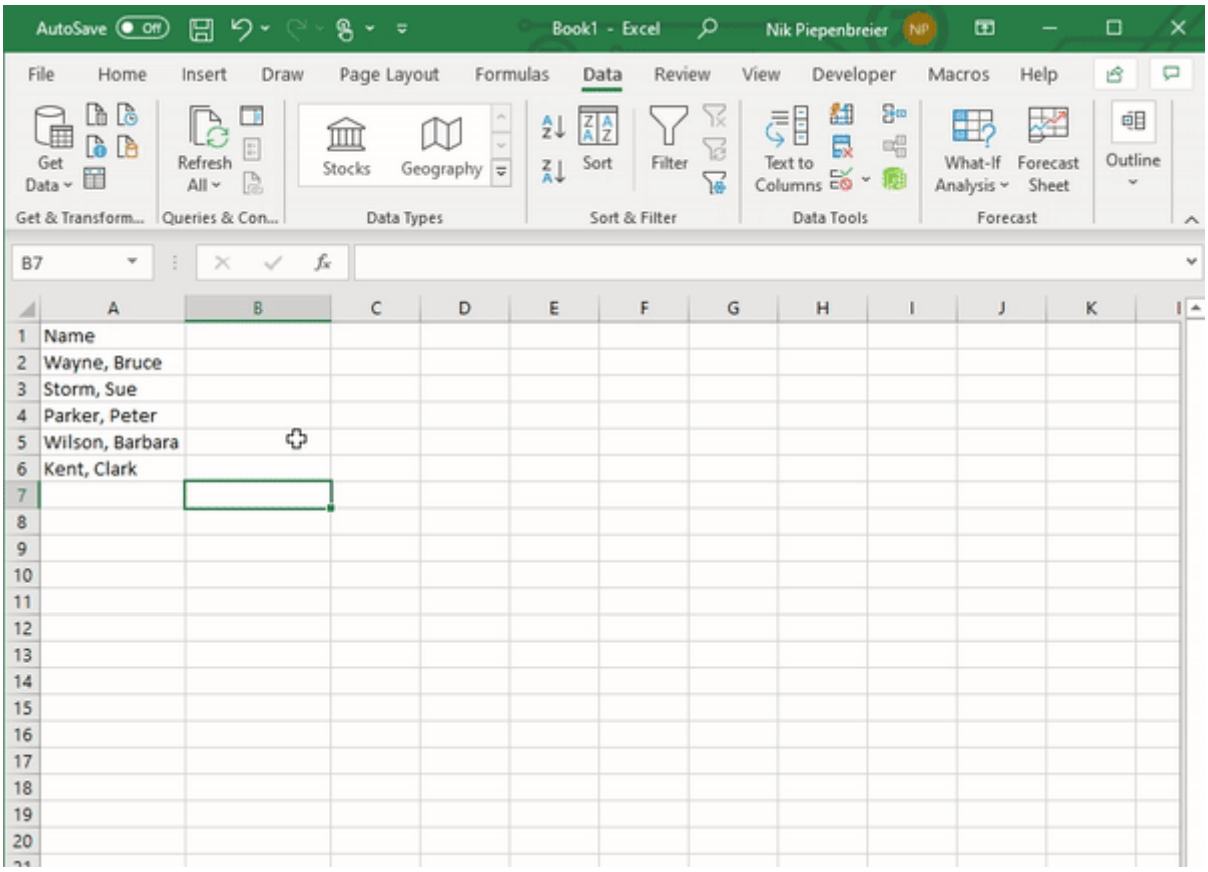
Are you wanting to dip your toes into the fun world of Python and Pandas?

Python is often regarded as a bit more challenging to pick up than Excel, but as having limitless potential. This tutorial takes you step-by-step through the functions you already use in Excel.

## Text-to-Columns in Pandas

Text to columns is one of those essential functions when you're given large datasets of weirdly structured data.

For example, you may be given a column of people's names. But instead of having one column for the last name and another for the first name, you have a single column of *Last Name, First Name*.




Text to Columns in Excel. Source: Nik Piepenbreier

Excel's text to column feature lets you easily split this data into separate columns.

You simply select the column, click Data → Text to Columns, and delimit by a comma.

And voila!

Now to do this in Pandas is just as easy!

Let's take a look at this code here:

```
import pandas as pd
df = pd.DataFrame({'Name': ['Wayne, Bruce', 'Storm, Sue', 'Parker, Peter', 'Wilson, Barbara',
print(df)

#Returns:
```

```
#Returns:
#              Name
#0     Wayne, Bruce
#1       Storm, Sue
#2    Parker, Peter
#3  Wilson, Barbara
#4      Kent, Clark
```

datagy-intermediate01.py hosted with ♡ by GitHub view raw

Text to columns in Pandas. Source: Nik Piepenbreier

To create two new columns with the last name and first names, simply write the following:

```
df[['Last Name', 'First Name']] = df['Name'].str.split(', ', expand=True)
print(df)

# Returns:
#              Name Last Name First Name
#0     Wayne, Bruce     Wayne      Bruce
#1       Storm, Sue     Storm        Sue
#2    Parker, Peter    Parker      Peter
#3  Wilson, Barbara    Wilson    Barbara
#4      Kent, Clark      Kent      Clark
```

datagy-intermediate02.py hosted with ♡ by GitHub view raw

Text to Columns in Pandas — code. Source: Nik Piepenbreier

Let's take a look at the split function:

- We specified that we wanted to split on the comma (,).
- We also indicated that expand=True, meaning that the data is split into new columns. If the value were False, a list of strings would be created.
- You can also use n= to specify the number of times you want to split a string. The default value is -1, meaning all instances.

## Replace Text Functions (e.g., RIGHT(), PROPER()) With Pandas

Excel is also jam-packed with different text functions. You can use functions like LEFT() and RIGHT() to extract characters, and PROPER() and LOWER() to modify text within cells.

Let's load another dataframe and see how these can be done with Pandas:

```
df2 = pd.DataFrame({
    'Postal Code': ['M1R0E9', 'V0S1G2', 'A1SB1G'],
    'Name': ['Nik', 'Mel', 'Jane']})
print(df2)

# Returns:
#   Postal Code  Name
#0      M1R0E9   Nik
#1      V0S1G2   Mel
#2      A1SB1G  Jane
```

datagy-intermediate03.py hosted with ♡ by GitHub view raw

Loading more data. Source: Nik Piepenbreier

Now, let's apply some of the text functions:

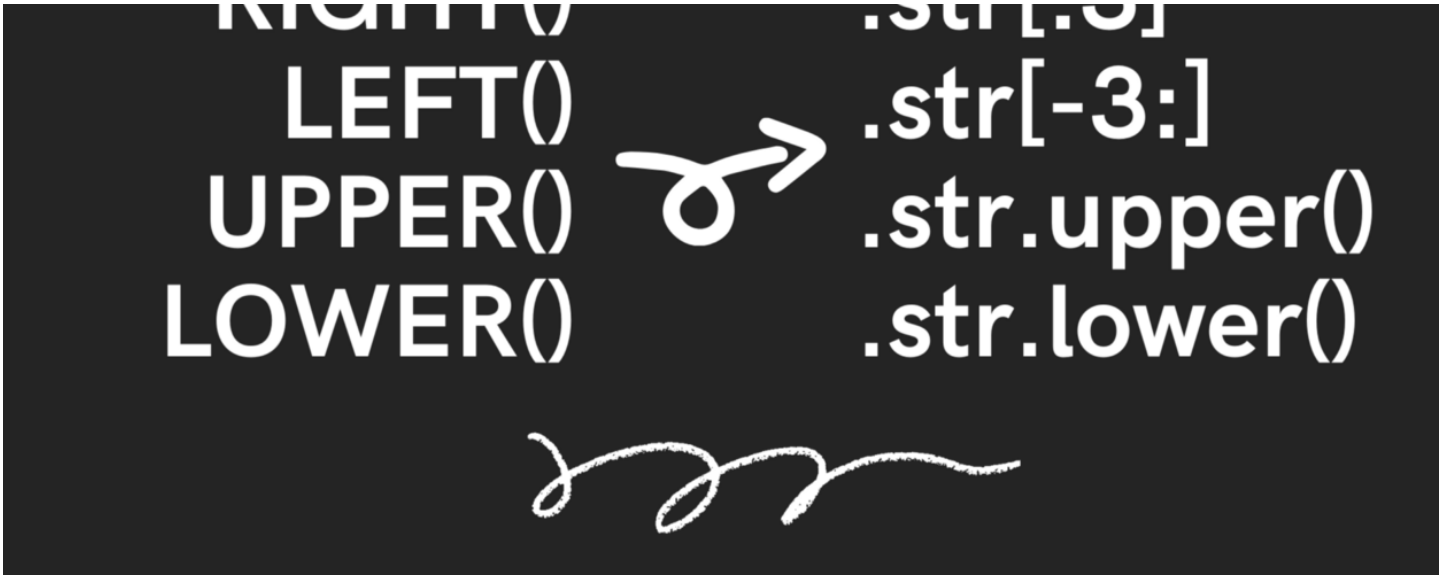
```
df2['Left'] = df22['Postal Code'].str[:3]  # LEFT() Equivalent
df2['Right'] = df['Postal Code'].str[-3:]  # RIGHT() Equivalent
df2['Lower'] = df2['Name'].str.lower()     # LOWER() Equivalent
df2['Upper'] = df2['Name'].str.upper()     # UPPER() Equivalent
print(df2)

# Returns:
#  Postal Code  Name Lower Upper Left Right
#0      M1R0E9   Nik   nik   NIK  M1R   0E9
#1      V0S1G2   Mel   mel   MEL  V0S   1G2
#2      A1SB1G  Jane  jane  JANE  A1S   B1G
```

datagy-intermediate04.py hosted with ♡ by GitHub view raw

Applying string functions. Source: Nik Piepenbreier

The LEFT() and RIGHT() functions work by indexing the strings in each column. As strings are *iterable* objects in Python, you can index them.

By writing [:3], you are asking Python to return the characters up to (but not including) the third one.




Some common text functions in Excel and in Pandas.Source: Nik Piepenbreier

## Apply Calculations Across Columns

Excel makes acting on columns with operations incredibly intuitive. As each cell has a specific reference, you can use that cell reference to easily multiply it by a value, add two cells together, etc.

It's not as immediately clear that you can do this in Pandas, too!

Let's load a sample dataframe that contains two columns of weights, which we'll assume are in kilograms:

```
df3 = pd.DataFrame({'weight1': [4,7,2], 'weight2':[6,3,9]})
print(df3)

# Returns
#   weight1  weight2
#0        4        6
#1        7        3
#2        2        9
```

datagy-intermediate05.py hosted with ♡ by GitHub view raw

Loading a third dataframe. Source: Nik Piepenbreier

If we wanted to create a new column that contains the sum of weight1 and weight2, we can simply add them together:

```
df3['sum'] = df3['weight1'] + df3['weight2']
print(df3)

```

```
# Returns:
#   weight1  weight2  sum
#0        4        6   10
#1        7        3   10
#2        2        9   11
```

datagy-intermediate06.py hosted with ♡ by GitHub view raw

Adding two columns. Source: Nik Piepenbreier

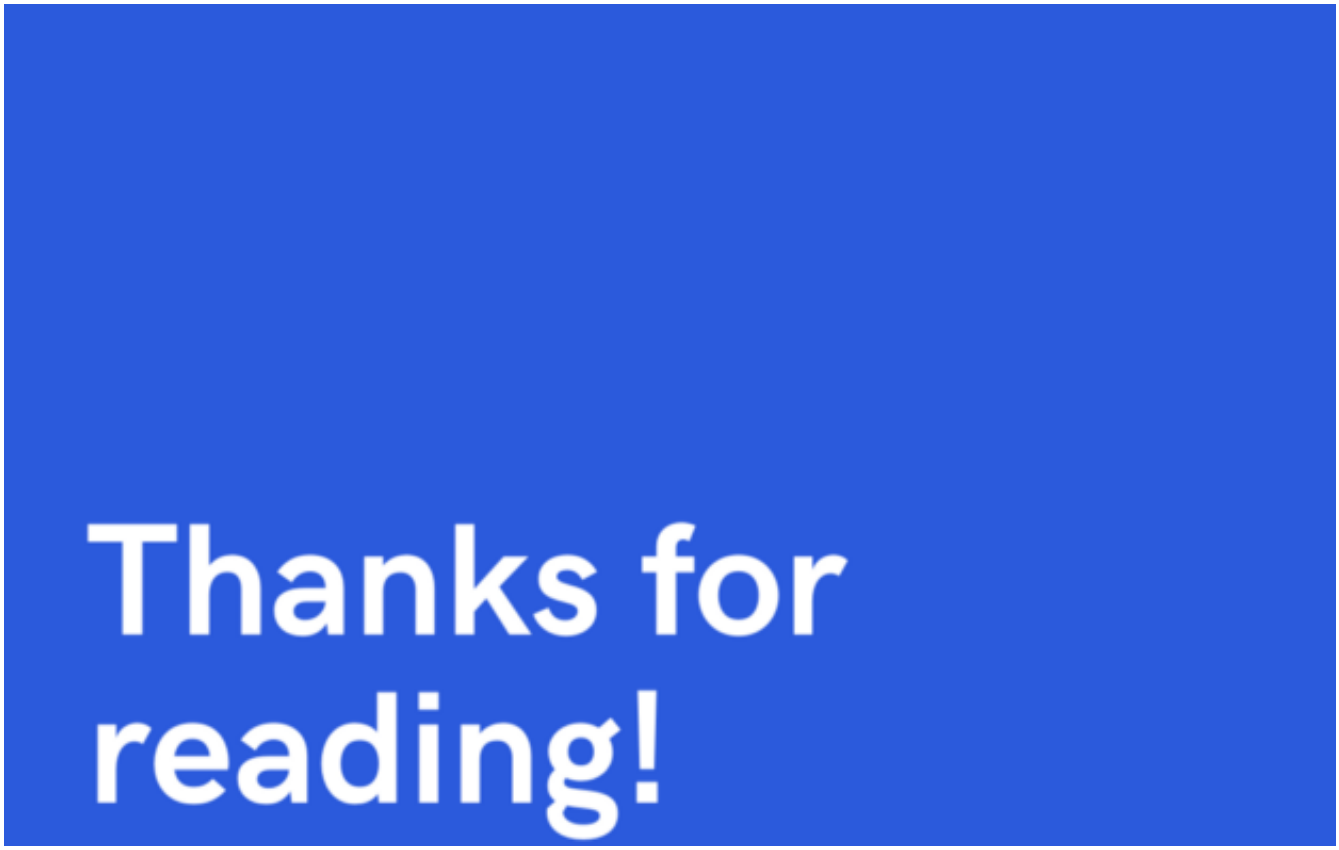
Finally, if we wanted to convert the sum column to pounds (by multiplying it by 2.2), you can apply that multiplication directly:

```
df3['pounds'] = df3['sum'] * 2.2
print(df3)

# Returns:
#   weight1  weight2  sum  pounds
#0        4        6   10    22.0
#1        7        3   10    22.0
#2        2        9   11    24.2
```

datagy-intermediate07.py hosted with ♡ by GitHub view raw

Multiplying a column in Pandas. Source: Nik Piepenbreier

## Conclusion



Thanks for reading! Source: Nik Piepenbreier

Thanks for reading this article! I hope you learned something and are feeling more comfortable dipping your toes into the fun world of Python and Pandas!

If you want to dive deeper into this, check out my other articles or my eBook that takes you from complete beginner all through intermediate Pandas! You can find that here: https://gumroad.com/l/introtopythonfordatascience

Data Science Python Programming Software Development Data