

Extracted from:

Pandas Brain Teasers

Exercise Your Mind

This PDF file contains pages extracted from *Pandas Brain Teasers*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2021 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

Pandas Brain Teasers

Exercise Your Mind



Miki Tebeka

edited by Margaret Eldridge

Pandas Brain Teasers

Exercise Your Mind

Miki Tebeka

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit <https://pragprog.com>.

The team that produced this book includes:

CEO: Dave Rankin

COO: Janet Furlow

Managing Editor: Tammy Coron

Development Editor: Margaret Eldridge

Copy Editor: Jennifer Whipple

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2021 The Pragmatic Programmers, LLC.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-901-4

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—September 2021

To all the data nerds out there, you rock!

Late Addition

archer.py

import pandas as pd

```
df = pd.DataFrame([
    ['Sterling', 83.4],
    ['Cheryl', 97.2],
    ['Lana', 13.2],
], columns=['name', 'sum'])
df.late_fee = 3.5
print(df)
```

Guess the Output



Try to guess what the output is before moving to the next page.

This code will print:

```

      name      sum
0  Sterling  83.4
1   Cheryl  97.2
2    Lana   13.2

```

Where did the `late_fee` column go?

Python's objects are very dynamic. You can add attributes to most of them as you please.

```

In [1]: class Point:
...:     def __init__(self, x, y):
...:         self.x, self.y = x, y
In [2]: p = Point(1, 2)
In [3]: p.x, p.y
Out[3]: (1, 2)
In [4]: p.z = 3
In [5]: p.z
Out[5]: 3

```

Pandas lets you access columns both by square brackets (e.g., `df[name]`) and by attribute (e.g., `df.name`). I recommend using square brackets at all times. One reason is, as we saw, when you add an attribute to a `DataFrame`, it does not register as a new column. Another reason is that column names in CSV, JSON, and other formats can contain spaces or other characters that are not valid Python identifiers, meaning you won't be able to access them with attribute access. `df.product id` will fail while `df['product id']` will work.

And the last reason is that it's confusing:

```

In [6]: df.sum
Out[6]:
<bound method DataFrame.sum of      name      sum
0  Sterling  83.4
1   Cheryl  97.2
2    Lana   13.2>

```

You get the `DataFrame` `sum` method and not the `sum` column. Also:

```

In [7]: df.late_fee
Out[7]: 3.5

```

You probably expected `late_fee` to be a `Series` like the other columns.

Sometimes you'd like to add metadata to a `DataFrame`, say, the name of the file the data was read from.

Instead of adding a new attribute, for example, `df.Originating_file = '/path/to/sales.db'`, there's an experimental attribute called `attrs` for storing metadata in a `DataFrame`.

```
In [8]: df.attrs['originating_file'] = '/path/to/sales.db'
```

```
In [9]: df.attrs
```

```
Out[9]: {'originating_file': '/path/to/sales.db'}
```

Further Reading

Indexing Basics in the Pandas Documentation

pandas.pydata.org/docs/user_guide/indexing.html#basics

Identifiers and Keywords in the Python Documentation

docs.python.org/3/reference/lexical_analysis.html#identifiers

DataFrame.attrs in the Pandas Documentation

pandas.pydata.org/docs/reference/api/pandas.DataFrame.attrs.html#pandas.DataFrame.attrs