

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!

Off with Their NaNs

`not_nan.py`

```
import numpy as np
import pandas as pd

s = pd.Series([1, np.nan, 3])
print(s[~(s == np.nan)])
```

Guess the Output



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

This code will print:

```
0    1.0
1    NaN
2    3.0
dtype: float64
```

We covered some of the floating-point oddities in the puzzle [Multiplying](#). NaN (or np.nan) is another oddity. The name NaN stands for *not a number*. It serves two purposes: illegal computation and missing values.

Here's an example of a bad computation:

```
In [1]: np.float64(0)/np.float64(0)
RuntimeWarning: invalid value encountered in \
double_scalars np.float64(0)/np.float64(0)
Out[1]: nan
```

You see a warning but not an exception, and the return value is nan.

nan does not equal any number, *including itself*.

```
In [2]: np.nan == np.nan
Out[2]: False
```

To check that a value is nan, you need to use a special function such as pandas.isnull:

```
In [3]: pd.isnull(np.nan)
Out[3]: True
```

You can use pandas.isnull to fix this teaser.

not_nan_fixed.py

```
import numpy as np
import pandas as pd

s = pd.Series([1, np.nan, 3])
print(s[~pd.isnull(s)])
```

pandas.isnull works with all Pandas “missing” values: None, pandas.NaT (not a time), and the new pandas.NA.

Floating points have several other special “numbers” such as inf (infinity), -inf, -0, +0, and others. You can learn more about them in the following links.

Further Reading

pandas.isnull in the Pandas Documentation

pandas.pydata.org/pandas-docs/stable/reference/api/pandas.isnull.html

Experimental NA Scalar to Denote Missing Values in the Pandas Documentation

pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html#missing-data-na

Floating-Point Arithmetic: Issues and Limitations in the Python Documentation

docs.python.org/3/tutorial/floatingpoint.html

floating point zine by Julia Evans

twitter.com/b0rk/status/986424989648936960

What Every Computer Scientist Should Know About Floating-Point Arithmetic

docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html