

check if variable is dataframe

Asked 6 years, 8 months ago Active 8 months ago Viewed 82k times

when my function f is called with a variable I want to check if var is a pandas dataframe:

104

```
def f(var):
    if var == pd.DataFrame():
        print "do stuff"
```

I guess the solution might be quite simple but even with

12

```
def f(var):
    if var.values != None:
        print "do stuff"
```

I can't get it to work like expected.

python

pandas

edited Dec 14 '18 at 7:24



Max Ghenis

5,826 6 39 69

asked Feb 11 '13 at 9:10



trbck

1,374 5 14 21

- 1 Your code says "if var is equal to an empty dataframe". What you really want is "if the type of var is equal to the type pd.DataFrame ". You can check that using `isinstance` – Katriel Feb 11 '13 at 9:17

2 Answers

Use [isinstance](#) , nothing else:

132

```
if isinstance(x, pd.DataFrame):
    ... # do something
```



[PEP8](#) says explicitly that `isinstance` is the preferred way to check types

```
No: type(x) is pd.DataFrame
No: type(x) == pd.DataFrame
Yes: isinstance(x, pd.DataFrame)
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
if obj.__class__.__name__ == 'DataFrame':
    expect_problems_some_day()
```

`isinstance` handles inheritance (see [What are the differences between `type\(\)` and `isinstance\(\)`?](#)). For example, it will tell you if a variable is a string (either `str` or `unicode`), because they derive from `basestring`)

```
if isinstance(obj, basestring):
    i_am_string(obj)
```

Specifically for `pandas DataFrame` objects:

```
import pandas as pd
isinstance(var, pd.DataFrame)
```

edited Jan 22 at 8:01



cs95

171k 34 253 315

answered Feb 11 '13 at 9:23



Jakub M.

14.5k 30 89 151

Use the built-in `isinstance()` function.

126

```
import pandas as pd
```

```
def f(var):
    if isinstance(var, pd.DataFrame):
        print("do stuff")
```

edited Jan 22 at 8:01



cs95

171k 34 253 315

answered Feb 11 '13 at 9:15



Rutger Kassies

33.4k 10 81 83

- 3 How can you generalise this to the case in which a user may use the function you define, but didn't `import pandas as pd`, but instead just `import pandas`? Just perform an `or` on both possibilities, or is there something more sophisticated I don't know of? – [n1k31t4](#) Jun 18 '17 at 22:14

A potential solution could be to put the import statement inside the function so there is no chance of a user importing `pandas` using some other method. To speed things up (to avoid importing the entire `panda` library for a simple check) you could just use something like `import pandas.DataFrame as panda_type` and then inside then check the array type using `isinstance(var, panda_type)` – [pacificgilly1992](#) Dec 31 '18 at 20:13