

A hacker's guide to debugging in Python

0800
1000

stopped - antan ✓

13" VC (032) MP - MC

1.582647000

9.037 846 995 correct

2.13047615 (3) 4615925059(-2)

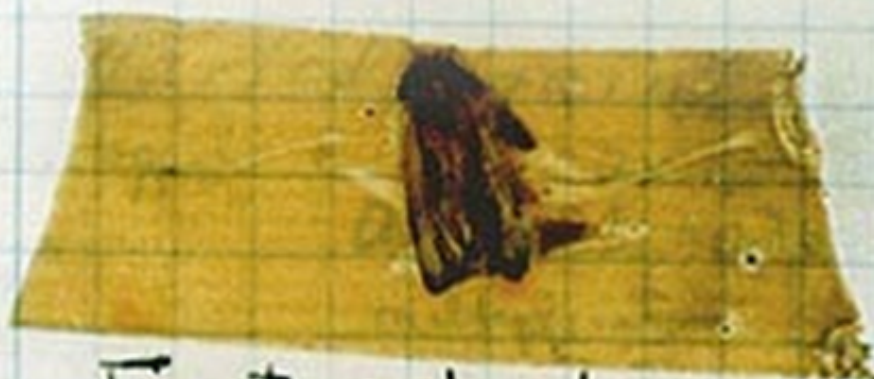
Relays 6-2 in 033 failed special speed test
in relay " 10.000 test.

Relays changed

1100
1525

Started Cosine Tape (Sine check)
Started Multi-Adder Test.

1545



Relay #70 Panel F
(moth) in relay.

1630

First actual case of bug being found.
Antan started.

1700

closed down.

Relay 2145
Relay 337

Things break.
So they should break...

Things break.
So they should break...

- **gracefully**

Things break.
So they should break...

- **gracefully**
- **helpfully**

Things break.

So they should break...

- gracefully
- helpfully
- silently

Exceptions

Some examples:

Exceptions

Some examples:

- **KeyError** - for when a key can't be found in a dictionary

Exceptions

Some examples:

- **KeyError** - for when a key can't be found in a dictionary
- **NameError** - for problems related to variable names

Exceptions

Some examples:

- **KeyError** - for when a key can't be found in a dictionary
- **NameError** - for problems related to variable names
- **TypeError** - for problems related to data types

Exceptions

Some examples:

- **KeyError** - for when a key can't be found in a dictionary
- **NameError** - for problems related to variable names
- **TypeError** - for problems related to data types
- ...etc. See all of the built-in ones [here](#).

Packages can define their own exceptions

...and you can too!

```
class MissingDataError(Exception):  
    """To be raised when some non-optional  
    data is missing."""  
    pass
```

Raising an exception

```
# every movie must have a title  
if 'title' not in mojo_dict.keys():  
    raise MissingDataError('JSON must have a'  
                           'key called "title"')
```

**Why spend so much time on
making different kinds of errors?**

Why spend so much time on making different kinds of errors?

- 1. Makes debugging easier**

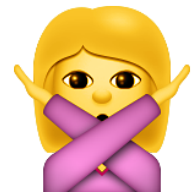


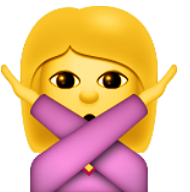
Why spend so much time on making different kinds of errors?

- 1. Makes debugging easier**
- 2. We can deal with them programmatically**

Catching an exception

```
try:
    year = int(meta_dict.get('year'))
except ValueError:
    logging.warning("No parseable year for movie %s.",
                    title)
    movie['year'] = None
```


except is not for making errors “go away”.

  **DON'T DO THIS**  

```
try:  
    process_data(thing)  
except:  
    pass
```

Reading a traceback

```

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5d138cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                     coerce_float=coerce_float,
-> 4876                                     dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993     if columns is None:
    4994         gen = (list(x.keys()) for x in data)
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997     # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dicts_to_arrays at 0x...>
    4992 def _list_of_dict_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
AttributeError: 'list' object has no attribute 'keys'

```

your function
call (where
the error
happened)

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-1-f5d138cdde43> in <module>()  
      1 import pandas as pd  
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]  
----> 3 pd.DataFrame(stuff)  
  
/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, index, columns, dtype, copy)  
    243         if len(data) > 0:  
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:  
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)  
    246                 columns = _ensure_index(columns)  
    247  
  
/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, coerce_float, dtype)  
    4874         return _list_of_dict_to_arrays(data, columns,  
    4875                                     coerce_float=coerce_float,  
-> 4876                                     dtype=dtype)  
    4877     elif isinstance(data[0], Series):  
    4878         return _list_of_series_to_arrays(data, columns,  
  
/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)  
    4993     if columns is None:  
    4994         gen = (list(x.keys()) for x in data)  
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)  
    4996  
    4997     # assure that they are of the base dict class and not of derived  
  
pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()  
  
/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dicts_to_arrays at 0x100000000>  
    4992 def _list_of_dicts_to_arrays(data, columns, coerce_float=False, dtype=None):  
    4993     if columns is None:  
-> 4994         gen = (list(x.keys()) for x in data)  
    4995         columns = lib.fast_unique_multiple_list_gen(gen)  
    4996  
  
AttributeError: 'list' object has no attribute 'keys'
```

the function
that function
called

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5d138cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, index, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                       coerce_float=coerce_float,
-> 4876                                       dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993         if columns is None:
    4994             gen = (list(x.keys()) for x in data)
-> 4995             columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997         # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dicts_to_arrays at 0x...>
    4992 def _list_of_dict_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996

AttributeError: 'list' object has no attribute 'keys'
```

etc...

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5d138cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                       coerce_float=coerce_float,
-> 4876                                       dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993     if columns is None:
    4994         gen = (list(x.keys()) for x in data)
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997     # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator>_list_of_dicts_to_arrays(data, columns, coerce_float=False, dtype=None)
    4992 def _list_of_dict_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996

AttributeError: 'list' object has no attribute 'keys'
```

etc...

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5d138cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                       coerce_float=coerce_float,
-> 4876                                       dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993     if columns is None:
    4994         gen = (list(x.keys()) for x in data)
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997     # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dicts_to_arrays at 0x...>
    4992 def _list_of_dict_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996

AttributeError: 'list' object has no attribute 'keys'
```


The line of
code that
actually
raised the
error

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5dl38cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, index, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                     coerce_float=coerce_float,
-> 4876                                     dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993     if columns is None:
    4994         gen = (list(x.keys()) for x in data)
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997     # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dict_to_arrays at 0x...>
    4992 def _list_of_dict_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996

AttributeError: 'list' object has no attribute 'keys'
```


The error
itself (its
type, and
any message
supplied)

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-1-f5d138cdde43> in <module>()
      1 import pandas as pd
      2 stuff = [{'a': 3, 'b': 1, 'c': 2}, {'a': 1, 'b': 2, 'c': 3}, ['a', 'b', 'c']]
----> 3 pd.DataFrame(stuff)

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in __init__(self, data, index, columns, dtype, copy)
    243         if len(data) > 0:
    244             if is_list_like(data[0]) and getattr(data[0], 'ndim', 1) == 1:
--> 245                 arrays, columns = _to_arrays(data, columns, dtype=dtype)
    246                 columns = _ensure_index(columns)
    247

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _to_arrays(data, columns, dtype, coerce_float, dtype)
    4874         return _list_of_dict_to_arrays(data, columns,
    4875                                     coerce_float=coerce_float,
-> 4876                                     dtype=dtype)
    4877     elif isinstance(data[0], Series):
    4878         return _list_of_series_to_arrays(data, columns,

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in _list_of_dicts_to_arrays(data, columns, coerce_float, dtype)
    4993     if columns is None:
    4994         gen = (list(x.keys()) for x in data)
-> 4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996
    4997     # assure that they are of the base dict class and not of derived

pandas/lib.pyx in pandas.lib.fast_unique_multiple_list_gen (pandas/lib.c:9054)()

/Users/brian/.virtualenvs/c1labs/lib/python2.7/site-packages/pandas/core/frame.pyc in <generator object _list_of_dicts_to_arrays at 0x...>
    4992 def _list_of_dicts_to_arrays(data, columns, coerce_float=False, dtype=None):
    4993     if columns is None:
-> 4994         gen = (list(x.keys()) for x in data)
    4995         columns = lib.fast_unique_multiple_list_gen(gen)
    4996

AttributeError: 'list' object has no attribute 'keys'
```

```
----> 3 pd.DataFrame(stuff)
```

Hmm, error occurs when I'm trying to make a new DataFrame...

```
AttributeError: 'list' object has no attribute 'keys'
```

Looks like somewhere deep in Pandas, it's trying to get keys from a list...

Which means it's probably expecting a dictionary and getting a list instead...

```
for thing in stuff:  
    print type(thing), thing
```

```
<type 'dict'> {'a': 3, 'c': 2, 'b': 1}
```

```
<type 'dict'> {'a': 1, 'c': 3, 'b': 2}
```

```
<type 'list'> ['a', 'b', 'c']
```

Yep. Oops.

pdb

```
import pdb
```

```
...
```

```
pdb.set_trace()
```


Logging



How do I know when this thing is done?
How do I know what happened?

A good way:

```
print "Done with %d of %d" % (i, total)
```

A slightly better way:

```
import logging

# will print a message to the console
logging.warning('Watch out!')

# will not print anything
logging.info('I told you so')
```


Levels of logging

- **DEBUG** - notes for you, while working on code
- **INFO** - notes that everything is going OK
- **WARNING** - something seems wrong but it's not urgent. running out of disk space, or data is probably too small, or things like that

- **ERROR** - something broke. things did not work.
- **CRITICAL** - oh 💩. things are seriously screwed. the whole program probably stopped running.

The most robust way:

```
import logging
logging.basicConfig(filename='example.log',
                    level=logging.DEBUG)
logging.debug('This message should go to the log file')
logging.info('So should this')
logging.warning('And this, too')
```

That's just the beginning.

You can customize the information it provides with things like dates, version numbers, policies for logging to multiple files, and more.

Python logging howto