

**You can't automatically
check if code is good**

Code quality in Python - tools and reasons

Radosław Ganczarek

Rice's theorem

Rice's theorem

(You probably haven't heard of it)



for any non-trivial property of partial functions, no general and effective method can decide whether an algorithm computes a partial function with that property

Rice's theorem

In the common language:



you can't write an algorithm, that will check in finite time if a program has some nontrivial property

**You can't write a dictionary
containing all good programs**

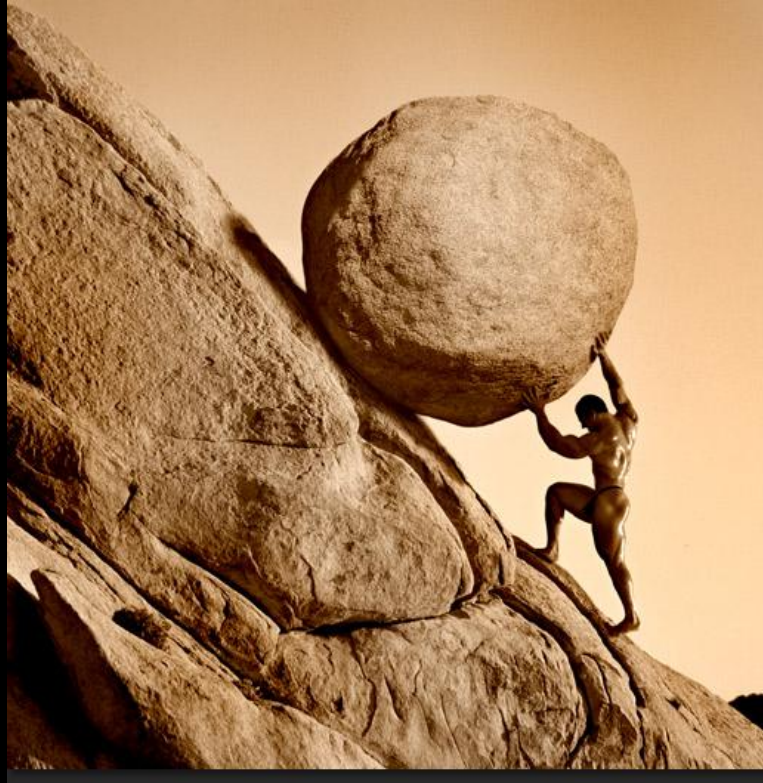


Examples

You can't automatically check if:

- program never fails
- there is any data set, for which program returns "1"
- program is well written and will be readable for humans
- program's WTF factor will be low

But you can try hard ...



Don't worry! It's not that hard!



We've got many tools

Checkers

Formatters

Import sorters

Coverage

Docstring checkers

Dead code checkers

...



What's this talk about?

YES

Developer's view

Checking code every day

General tools

Python 2.7

Jenkins

NO

Manager's view

Charts and tables

Django, flask, web2py etc.

Python 3.*

Buildbot, Travis etc.

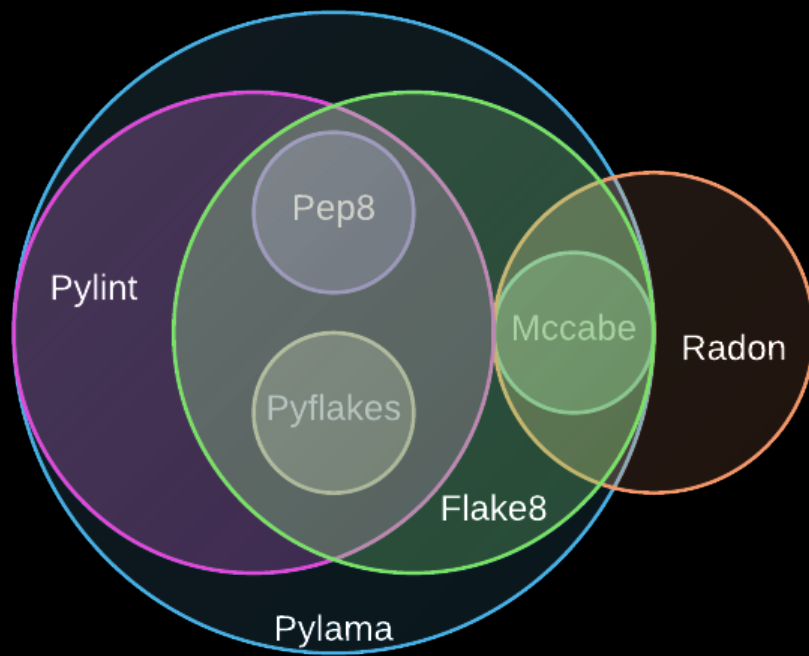
Checkers for other languages

Import graph visualisation

Let's get started!

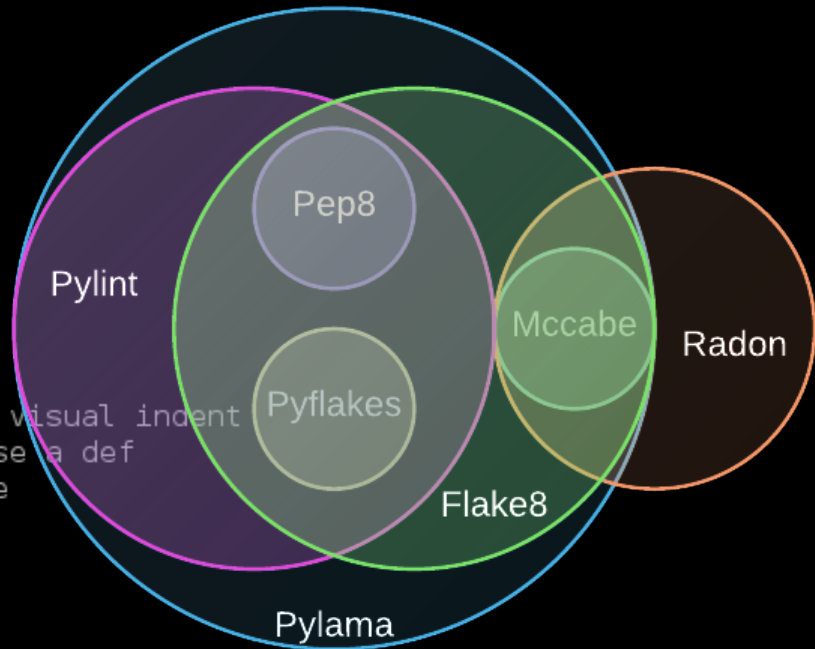


Checkers



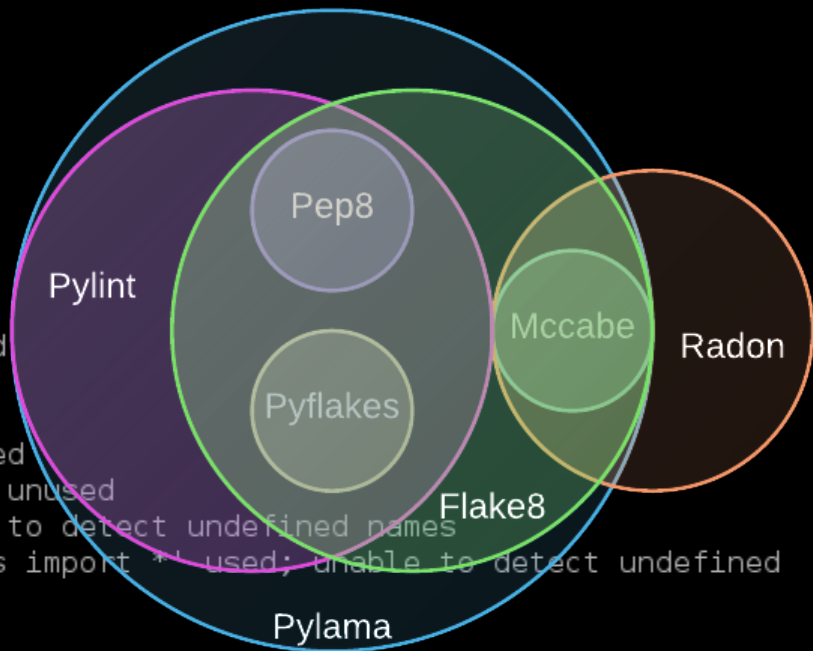
pep8

```
...py:36:13: W503 line break before binary operator
...py:150:80: E501 line too long (85 > 79 characters)
...py:86:17: E128 continuation line under-indented for visual indent
...py:215:9: E731 do not assign a lambda expression, use a def
...py:62:1: E402 module level import not at top of file
...py:182:20: W601 .has_key() is deprecated, use 'in'
...py:32:9: E301 expected 1 blank line, found 0
...py:35:1: E303 too many blank lines (3)
...py:160:14: E203 whitespace before ':'
...py:245:18: E127 continuation line over-indented for visual indent
...py:263:83: E721 do not compare types, use 'isinstance()'
...py:307:55: E221 multiple spaces before operator
...py:445:17: E265 block comment should start with '#'
...py:492:17: E125 continuation line with same indent as next logical line
...py:27:26: E251 unexpected spaces around keyword / parameter equals
...py:91:40: E261 at least two spaces before inline comment
...py:708:17: E129 visually indented line with same indent as next logical line
...py:771:1: E302 expected 2 blank lines, found 1
...py:841:1: E266 too many leading '#' for block comment
```



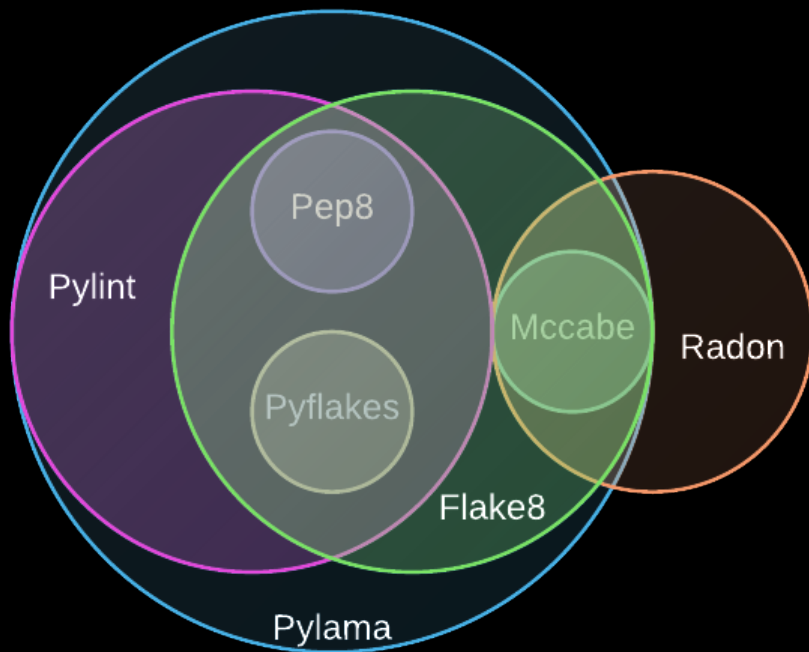
pyflakes

```
...py:13: 'GenericStackedInline' imported but unused
...py:13: 'GenericInlineModelAdmin' imported but unused
...py:16: 'GenericForeignKey' imported but unused
...py:16: 'GenericRelation' imported but unused
...py:19: 'BaseGenericInlineFormSet' imported but unused
...py:19: 'generic_inlineformset_factory' imported but unused
...py:4: 'from django.db.models import *' used; unable to detect undefined names
...py:15: 'from django.contrib.gis.db.models.aggregates import *' used; unable to detect undefined names
...py:18: 'GeoManager' imported but unused
...py:21: 'LineStringField' imported but unused
...py:21: 'MultiPolygonField' imported but unused
...py:21: 'MultiLineStringField' imported but unused
...py:21: 'PolygonField' imported but unused
...py:21: 'GeometryCollectionField' imported but unused
...py:21: 'PointField' imported but unused
...py:21: 'MultiPointField' imported but unused
...py:21: 'GeometryField' imported but unused
...py:2: 'from django.db.models.sql.aggregates import *' used; unable to detect undefined names
...py:10: 'from django.contrib.gis.db.backends.utils import *' used; unable to detect undefined nam
```



mccabe

```
("35:1: 'TestFieldType.test_custom_fields'", 1)
("23:1: 'TestFieldType.test_field_name'", 1)
("20:1: 'TestFieldType.setUp'", 1)
("29:1: 'TestFieldType.test_builtin_fields'", 1)
("184:1: 'ModelDetailView.get_context_data'", 18)
("63:1: 'TemplateTagIndexView.get_context_data'", 7)
("129:1: 'ViewIndexView.get_context_data'", 2)
("398:1: 'simplify_regex'", 2)
("358:1: 'get_readable_field_data_type'", 1)
("295:1: 'TemplateDetailView.get_context_data'", 4)
("34:1: 'BaseAdminDocsView.dispatch'", 2)
("324:1: 'load_all_installed_template_libraries'", 8)
("51:1: 'BookmarkletsView.get_context_data'", 1)
("175:1: 'ModelIndexView.get_context_data'", 1)
("366:1: 'extract_views_from_urlpatterns'", 8)
("148:1: 'ViewDetailView.get_context_data'", 5)
("96:1: 'TemplateFilterIndexView.get_context_data'", 7)
("42:1: 'BaseAdminDocsView.get_context_data'", 1)
("348:1: 'get_return_data_type'", 4)
("9:1: 'XViewMiddleware.process_view'", 2)
```



radon

django/db/models/aggregates.py

LOC: 162

LLOC: 91

SLOC: 127

Comments: 2

Multi: 21

Blank: 35

- Comment Stats

(C % L): 1%

(C % S): 2%

(C + M % L): 14%

django/db/models/query.py

LOC: 1770

LLOC: 1057

SLOC: 1565

Comments: 177

Multi: 345

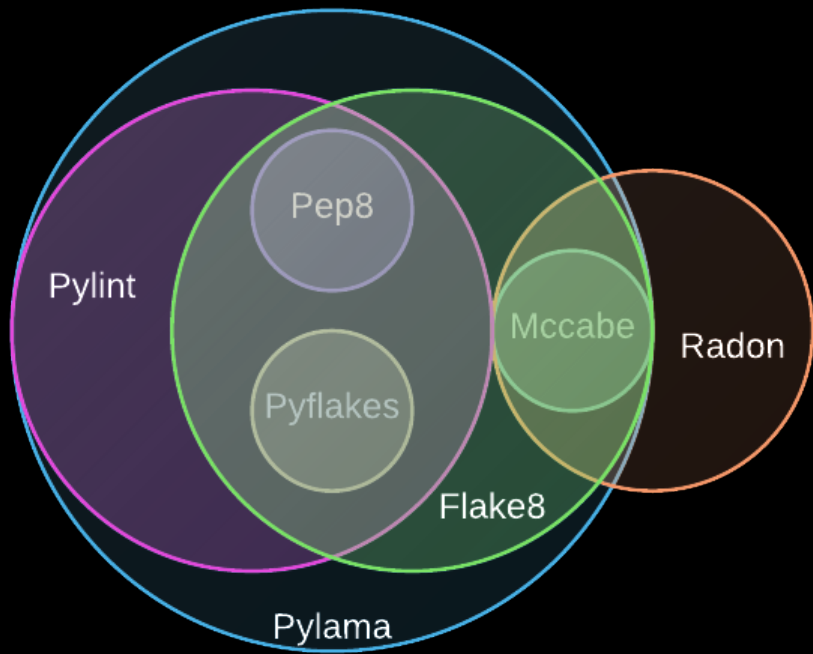
Blank: 205

- Comment Stats

(C % L): 10%

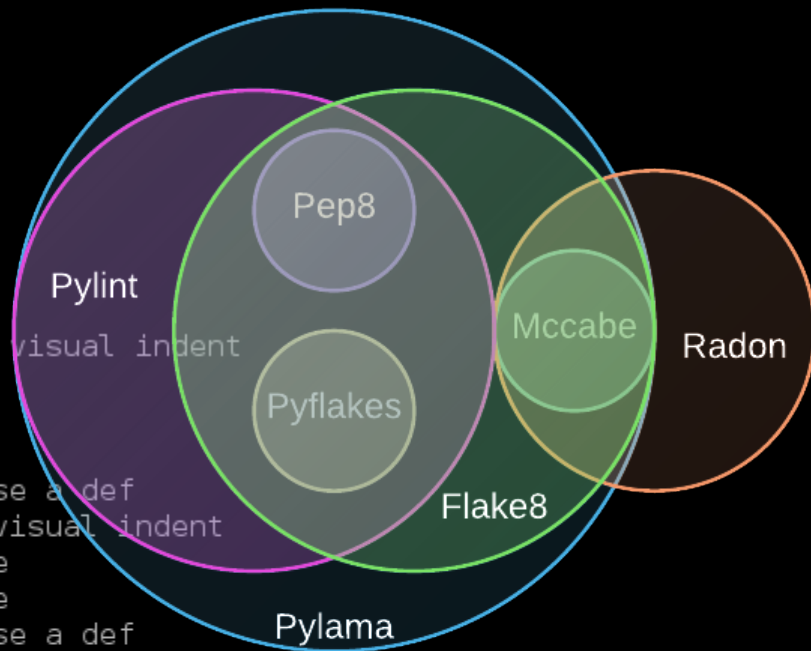
(C % S): 11%

(C + M % L): 29%



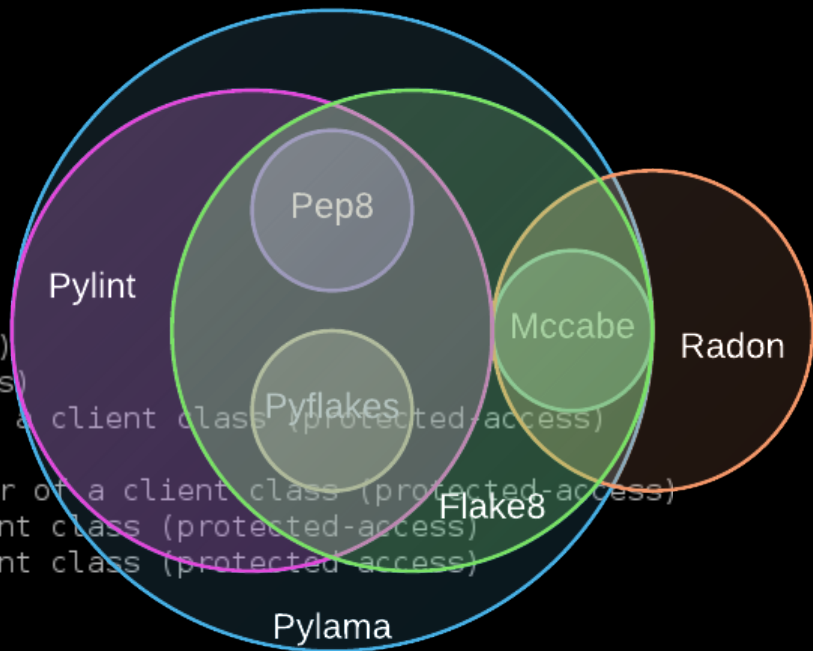
flake8

```
...py:7:80: E501 line too long (88 > 79 characters)
...py:86:17: E128 continuation line under-indented for visual indent
...py:36:13: W503 line break before binary operator
...py:150:80: E501 line too long (85 > 79 characters)
...py:241:29: W503 line break before binary operator
...py:215:9: E731 do not assign a lambda expression, use a def
...py:11:9: E128 continuation line under-indented for visual indent
...py:62:1: E402 module level import not at top of file
...py:85:1: E402 module level import not at top of file
...py:144:9: E731 do not assign a lambda expression, use a def
...py:182:20: W601 .has_key() is deprecated, use 'in'
...py:31:16: W601 .has_key() is deprecated, use 'in'
...py:32:9: E301 expected 1 blank line, found 0
...py:35:1: E303 too many blank lines (3)
...py:160:14: E203 whitespace before ':'
...py:245:18: E127 continuation line over-indented for visual indent
...py:263:83: E721 do not compare types, use 'isinstance()'
...py:307:55: E221 multiple spaces before operator
...py:445:17: E265 block comment should start with '#'
...py:492:17: E125 continuation line with same indent as next logical line
```



pylint

```
***** Module django.shortcuts
R: 27, 0: Too many arguments (8/5) (too-many-arguments)
R: 50, 0: Too many arguments (10/5) (too-many-arguments)
W: 85,16: Access to a protected member _current_app of a client class (protected-access)
C: 94, 0: Invalid argument name "to" (invalid-name)
W:131,18: Access to a protected member _default_manager of a client class (protected-access)
W:157,57: Access to a protected member _meta of a client class (protected-access)
W:171,57: Access to a protected member _meta of a client class (protected-access)
C:175, 0: Invalid argument name "to" (invalid-name)
***** Module django.contrib.admindocs.urls
W: 26, 0: Anomalous backslash in string: '\.'. String constant might be missing an r prefix.
lous-backslash-in-string)
W: 26, 0: Anomalous backslash in string: '\.'. String constant might be missing an r prefix.
lous-backslash-in-string)
C:  1, 0: Missing module docstring (missing-docstring)
C:  4, 0: Invalid constant name "urlpatterns" (invalid-name)
***** Module django.contrib.admindocs.views
C:110, 0: Line too long (104/100) (line-too-long)
C:135, 0: Line too long (109/100) (line-too-long)
C:240, 0: Line too long (106/100) (line-too-long)
```



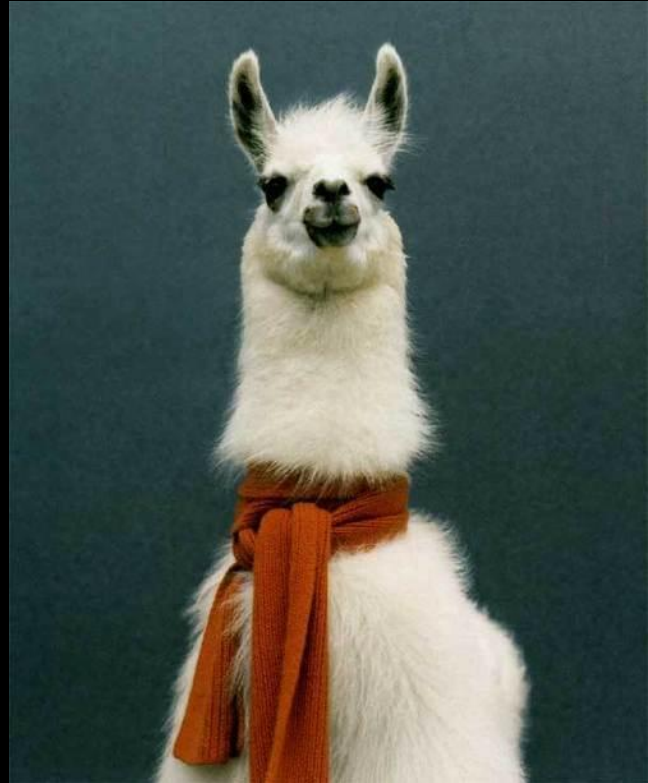
The lure of Pylama

Pro:

- all tools in one module

Con:

- Configuration



Other tools

pep257
vulture
isort



pep257

```
...py:1 at module level:
D100: Missing docstring in public module
...py:8 in public function `setup`:
D400: First line should end with a period (not 'e')
...py:8 in public function `setup`:
D205: 1 blank line required between summary line and description (found 0)
...py:1 at module level:
D400: First line should end with a period (not 's')
...py:1 at module level:
D205: 1 blank line required between summary line and description (found 0)
...py:27 in public function `render_to_response`:
D401: First line should be in imperative mood ('Return', not 'Returns')
...py:27 in public function `render_to_response`:
D400: First line should end with a period (not 'g')
...py:27 in public function `render_to_response`:
D205: 1 blank line required between summary line and description (found 0)
...py:50 in public function `render`:
D401: First line should be in imperative mood ('Return', not 'Returns')
...py:50 in public function `render`:
D400: First line should end with a period (not 'g')
```

Vulture

```
...py:100: Unused function 'csrf_failure'  
...py:87: Unused function 'technical_500_response'  
...py:134: Unused class 'SafeExceptionReporterFilter'  
...py:512: Unused function 'format_exception'  
...py:525: Unused function 'technical_404_response'  
...py:10: Unused function 'cache_page'  
...py:39: Unused function 'cache_control'  
...py:50: Unused function 'never_cache'  
...py:6: Unused function 'xframe_options_deny'  
...py:27: Unused function 'xframe_options_sameorigin'  
...py:48: Unused function 'xframe_options_exempt'  
...py:62: Unused attribute 'xframe_options_exempt'  
...py:8: Unused attribute '__doc__'  
...py:18: Unused function '_reject'  
...py:50: Unused function 'csrf_exempt'  
...py:59: Unused attribute 'csrf_exempt'  
...py:33: Unused attribute 'sensitive_variables'  
...py:73: Unused attribute 'sensitive_post_parameters'  
...py:18: Unused variable 'conditional_page'  
...py:173: Unused function 'etag'
```

isort

```
ERROR: shortcuts.py Imports are incorrectly sorted.
ERROR: contrib/admindocs/views.py Imports are incorrectly sorted.
ERROR: contrib/staticfiles/storage.py Imports are incorrectly sorted.
ERROR: contrib/staticfiles/finders.py Imports are incorrectly sorted.
ERROR: contrib/humanize/templatetags/humanize.py Imports are incorrectly sorted.
ERROR: contrib/sessions/backends/base.py Imports are incorrectly sorted.
ERROR: contrib/sessions/backends/file.py Imports are incorrectly sorted.
ERROR: contrib/contenttypes/admin.py Imports are incorrectly sorted.
ERROR: contrib/flatpages/forms.py Imports are incorrectly sorted.
ERROR: contrib/gis/maps/google/overlays.py Imports are incorrectly sorted.
ERROR: contrib/gis/maps/google/gmap.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/models/query.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/backends/oracle/adaptor.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/backends/oracle/introspection.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/backends/spatialite/base.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/backends/spatialite/introspection.py Imports are incorrectly sorted.
ERROR: contrib/gis/db/backends/mysql/introspection.py Imports are incorrectly sorted.
ERROR: contrib/gis/geometry/backend/geos.py Imports are incorrectly sorted.
ERROR: contrib/gis/gdal/feature.py Imports are incorrectly sorted.
ERROR: contrib/gis/gdal/geometries.py Imports are incorrectly sorted.
```

Extensions

flake8 (other checks - imports, docstrings etc.)

pylint (specific settings - Django, flask etc.)

pylama (new checkers)

browsing raw code vs AST

Extensions

Example: mccabe extension to flake8

```
def run(self):
    if self.max_complexity < 0:
        return
    visitor = PathGraphingAstVisitor()
    visitor.preorder(self.tree, visitor)
    for graph in visitor.graphs.values():
        if graph.complexity() >= self.max_complexity:
            text = self._error_tmpl % (graph.entity, graph.complexity())
            yield graph.lineno, 0, text, type(self)
```

Formatters

autopep8

yapf

isort



Autopep8

- Doesn't fix everything
- Leaves ugly line breaks

```
self.has_visible_field = not all(field in self.form.fields and  
                                self.form.fields[  
                                    field].widget.is_hidden  
                                for field in self.fields)
```



Yapf - our saviour

```
class Fieldline(object):
    def __init__(self, form, field, readonly_fields=None, model_admin=None):
        self.form = form # A django.forms.Form instance
        if not hasattr(field, "__iter__") or isinstance(field, six.text_type):
            self.fields = [field]
        else:
            self.fields = field
        self.has_visible_field = not all(
            field in self.form.fields and
            self.form.fields[field].widget.is_hidden for field in self.fields)
        self.model_admin = model_admin
        if readonly_fields is None:
            readonly_fields = ()
        self.readonly_fields = readonly_fields
```

Isort

```
from django import forms
from django.conf import settings
from django.contrib.admin.templatetags.admin_static import static
from django.contrib.admin.utils import (display_for_field, flatten_fieldsets,
                                         help_text_for_field, label_for_field,
                                         lookup_field)

from django.core.exceptions import ObjectDoesNotExist
from django.db.models.fields.related import ManyToManyRel
from django.forms.utils import flatatt
from django.template.defaultfilters import capfirst, linebreaksbr
from django.utils import six
from django.utils.deprecation import RemovedInDjango110Warning
from django.utils.encoding import force_text, smart_text
from django.utils.functional import cached_property
from django.utils.html import conditional_escape, format_html
from django.utils.safestring import mark_safe
from django.utils.translation import ugettext_lazy as _
```

Test coverage

coverage module

How to run

Common problems

Comparing with master



Utils

diff-cover

diff-quality

git-lint

hgdifffmetrics

scspell3k



diff-cover, diff-quality

```
-----  
Diff Coverage
```

```
Diff: origin/master...HEAD, staged, and unstaged changes
```

```
-----  
diff_cover/violations_reporter.py (50.0%): Missing lines 15
```

```
-----  
Total: 2 lines
```

```
Missing: 1 line
```

```
Coverage: 50%
```

```
-----  
Diff Quality
```

```
Quality Report: pylint
```

```
Diff: origin/master...HEAD, staged, and unstaged changes
```

```
-----  
diff_cover/tool.py (0.0%):
```

```
  70: C0303: (trailing-whitespace), : Trailing whitespace
```

```
diff_cover/violations_reporter.py (100%)
```

```
-----  
Total: 3 lines
```

```
Violations: 1 line
```

```
% Quality: 66%
```

git-lint, hgdiffmetrics

Linting file: **diff_cover/tool.py**

line 23, col 0: Warning: [unused-import]: Unused import django

line 48, col 0: Convention: [invalid-name]: parse_coverage_args: Invalid argument name "a"

line 48, col 30: Warning: [unused-argument]: parse_coverage_args: Unused argument 'a'

line 62, col 4: Warning: [global-variable-not-assigned]: parse_coverage_args: Using global for 'logging' but no assignment is done

line 62, col 4: Convention: [invalid-name]: parse_coverage_args: Invalid constant name "logging"

line 72, col 0: Convention: [trailing-whitespace]: Trailing whitespace

line 72, col 1: [W293]: blank line contains whitespace

scspell3k

```
'absolute' not found in dictionary (from token 'absolute')
'abstractmethod' not found in dictionary (from token 'abstractmethod')
'arabic' not found in dictionary (from token 'snippet_arabic_output')
'arabic' not found in dictionary (from token 'test_load_declared_arabic')
'argparse' not found in dictionary (from token 'argparse')
'changeset' not found in dictionary (from token 'changeset')
'chdir' not found in dictionary (from token 'chdir')
'classmethod' not found in dictionary (from token 'classmethod')
'clazz' not found in dictionary (from token 'clazz')
'cobertura' not found in dictionary (from token 'cobertura')
'cobertura' not found in dictionary (from token 'Cobertura')
'commandline' not found in dictionary (from token 'commandline')
'contextlib' not found in dictionary (from token 'contextlib')
'contextmanager' not found in dictionary (from token 'contextmanager')
'cssclass' not found in dictionary (from token 'cssclass')
'daeafb' not found in dictionary (from token '9daeafb')
'defaultdict' not found in dictionary (from token 'defaultdict')
'defs' not found in dictionary (from token 'get_style_defs')
'defs' not found in dictionary (from token 'style_defs')
'defs' not found in dictionary (from token 'test style defs')
```

Automation

tox

py.test

Jenkins

Github PR builder plugin



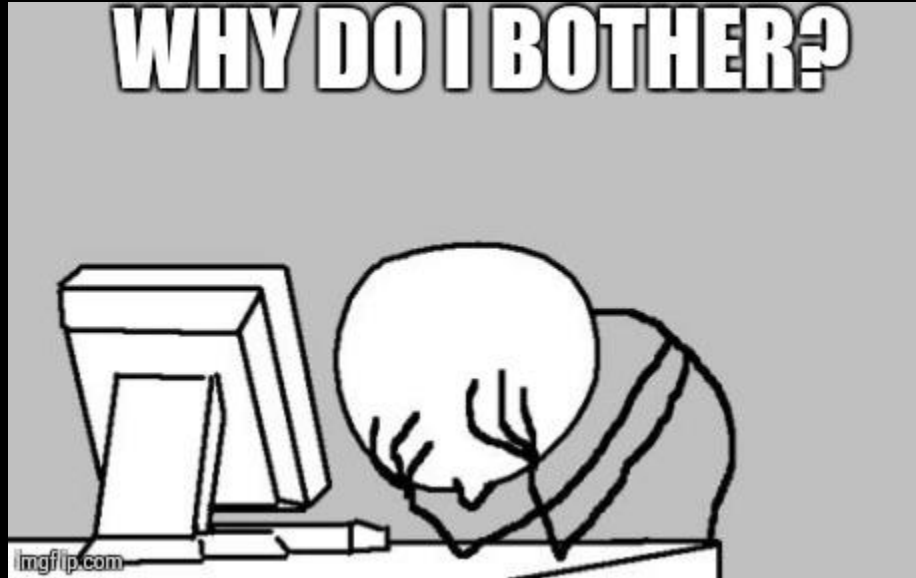
Tox example (from diff-cover app)

```
[tox]
envlist = py26, py27, py33, py34, pypy

[testenv]
whitelist_externals =
    /usr/bin/git
deps =
    -r{toxinidir}/requirements/test-requirements-py27-py3.txt
commands =
    coverage run -m nose {posargs}
    coverage xml
    git fetch origin master:refs/remotes/origin/master
    diff-cover coverage.xml
    diff-quality --violation=pep8
    diff-quality --violation=pyflakes
    diff-quality --violation=pylint

[testenv:py26]
deps =
    -r{toxinidir}/requirements/test-requirements-py26.txt
    argparse
```

OK, got it. But why?



Benefits?

- refactor confidence
- uniform style

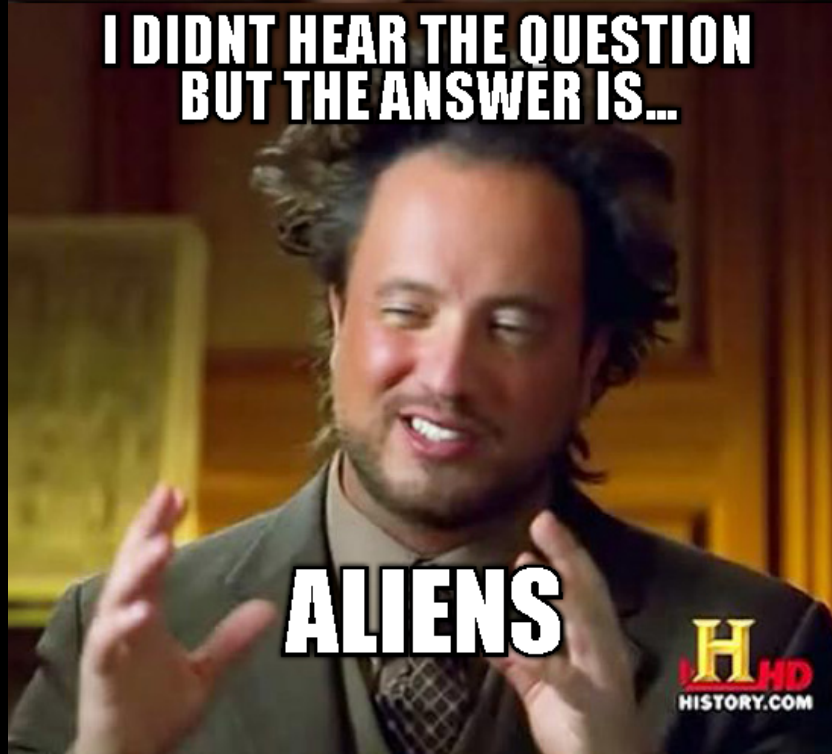
Pro tip: buzzwords for managers:

- maintainability
- readability
- extendability



Questions?

**I DIDNT HEAR THE QUESTION
BUT THE ANSWER IS...**



ALIENS

HHD
HISTORY.COM

Mandatory Monty Python screen



About me

Radosław Ganczarek

Senior Python Developer @ **STX**NEXT

(<http://stxnext.com>)

From Wrocław (Poland)

<https://pl.linkedin.com/in/dreamwalker>