Python Syntax-Checker Codes

42cc-pystyle

42cc1	Test has no docstring
42cc2	Function should be shorter than %d statements
42cc3	Function should be longer than a single if
42cc4	Commented out code

flake8-blind-except

B901	blind except: statement

flake8-coding

C101	Coding magic comment not found
C102	Unknown encoding found in coding magic comment

flake8_copyright

C801	Copyright notice not present

flake8-debugger

T002	{1} for {2} found as {3}

flake8-deprecated

D001	found {0:s} replace it with {1:s}

flake8_docstrings
(plugin wrapper to use pep257)

D100	Public module missing
D101	Public class missing
D102	Public method missing
D103	Public function missing
D200	One-line docstrings should fit on one line with quotes
D201	No blank lines allowed before docstring
D202	No blank lines allowed after docstring
D203	1 blank required before class docstring
D204	1 blank required after class docstring
D205	Blank line required between one-line summary and description
D206	Docstring should be indented with spaces, not tabs
D207	Docstring under-indented
D208	Docstring over-indented
D209	Put multi-line docstring closing quotes on separate line
D300	Use """triple double quotes"""
D301	Use r""" if any backslashes in your docstring
D302	Use u""" for Unicode docstrings (Python 2 only)
D400	First line should end with a period
D401	First line should be in imperative mood
D402	First line should not be the function's "signature"

flake8-double-quotes

Q000	Remove Single quotes

flake8-exact-pin

P001	exact pin found in install_requires

flake8-future

F481	Expected these __future__ imports, but got none: {0}

flake8-future-import

FI10	__future__ import "division" missing
FI11	__future__ import "absolute_import" missing
FI12	__future__ import "with_statement" missing
FI13	__future__ import "print_function" missing
FI14	__future__ import "unicode_literals" missing
FI15	__future__ import "generator_stop" missing
FI50	__future__ import "division" present
FI51	__future__ import "absolute_import" present
FI52	__future__ import "with_statement" present
FI53	__future__ import "print_function" present
FI54	__future__ import "unicode_literals" present
FI55	__future__ import "generator_stop" present

flake8-import-order

I100	Your import statements are in the wrong order
I101	The names in your from import are in the wrong order
I201	Missing newline between sections or imports

flake8-isort

I001	found unsorted imports

flake8-pep257

D100	Missing docstring in public module
D101	Missing docstring in public class
D102	Missing docstring in public method
D103	Missing docstring in public function
D104	Missing docstring in public package
D105	Missing docstring in magic method
D200	One-line docstring should fit on one line with quotes
D201	No blank lines allowed before function docstring
D202	No blank lines allowed after function docstring
D203	1 blank line required before class docstring
D204	1 blank line required after class docstring
D205	1 blank line required between summary line and description
D206	Docstring should be indented with spaces, not tabs
D207	Docstring is under-indented
D208	Docstring is over-indented
D209	Multi-line docstring closing quotes should be on a separate line
D210	No whitespaces allowed surrounding docstring text
D211	No blank lines allowed before class docstring
D300	Use """triple double quotes"""
D301	Use r""" if any backslashes in a docstring
D302	Use u""" for Unicode docstrings
D400	First line should end with a period
D401	First line should be in imperative mood
D402	First line should not be the function's "signature"

flake8-pep3101

S001	found {0:s} formatter

flake8_plone_hasattr

P002	found "hasattr", consider replacing it

flake8-print

T001	print statement found

flake8-quotes

Q000	Remove bad quotes

flake8-regex

R***	Regex warnings

flake8-single-quotes

Q000	Strings should use single quotes

flake8_strict

S100	First argument on the same line
S101	Multi-line construct missing trailing comma

flake8-string-format

P101	format string does contain unindexed parameters
P102	docstring does contain unindexed parameters
P103	other string does contain unindexed parameters
P201	format call uses to large index (INDEX)
P202	format call uses missing keyword (KEYWORD)
P203	format call uses keyword arguments but no named entries
P204	format call uses variable arguments but no numbered entries
P205	format call uses implicit and explicit indexes together
P301	format call provides unused index (INDEX)
P302	format call provides unused keyword (KEYWORD)

flake8-snippets

S100	Snippet found: "{}"

flake8_tuple

T801	one element tuple

mccabe

C901	x is too complex (#)

ebb-lint

L100	docstring-related errors
L101	__init__ doesn't need a docstring; standard sphinx convention is to use the class-level docstring instead.
L102	tried to validate the docstring: {parse_error}
L103	don't start test docstrings with {prefix!r}
L104	use sphinx.ext.napoleon instead of reST fields like :{field}:
L200	errors related to dubious syntax
L201	container literals must have a trailing comma following the last element
L202	print is only allowed while debugging; use logging in real code
L203	debuggers are only allowed while debugging; make sure to remove this before committing
L204	implicit string literal concatenation requires all string literals to be enclosed in parentheses containing nothing but string literals
L205	do not define functions or classes in __init__.py
L206	do not use implicit relative imports
L207	'pass' is only necessary in non-optional suites containing no other statements
L208	be more specific about which exception types to silence
L209	{stmt} is not a function; parentheses are not required
L210	use a for loop instead of using {thing} for its side effect
L211	use a comprehension instead of {func} with lambda
L212	@staticmethod is always wrong; you probably want @classmethod
L300	errors related to formatting
L301	files must end with a trailing newline
L302	line too long ({length} > {which_limit} limit of {limit} characters{extra})
L303	noqa is ignored, so this comment is not useful

Python Syntax-Checker Codes

pep8-naming

N801   class names should use CapWords convention
N802   function name should be lowercase
N803   argument name should be lowercase
N804   first argument of a classmethod should be named 'cls'
N805   first argument of a method should be named 'self'
N806   variable in function should be lowercase
N811   constant imported as non constant
N812   lowercase imported as non lowercase
N813   camelcase imported as lowercase
N814   camelcase imported as constant

pep8 (errors)

E101   indentation has spaces & tabs
E111   indentation is not a multiple of four
E112   expected an indented block
E113   unexpected indentation
E114   indentation is not a multiple of four (comment)
E115   expected an indented block (comment)
E116   unexpected indentation (comment)
E121   continuation line under-indented for hanging indent
E122   continuation line missing indentation or outdented
E123   closing bracket does not match indentation of opening bracket's line
E124   closing bracket does not match visual indentation
E125   continuation line with same indent as next logical line
E126   continuation line over-indented for hanging indent
E127   continuation line over-indented for visual indent
E128   continuation line under-indented for visual indent
E129   visually indented line with same indent as next logical line
E131   continuation line unaligned for hanging indent
E133   closing bracket is missing indentation
E201   whitespace after '('
E202   whitespace before ')'
E203   whitespace before ':'
E211   whitespace before '('
E221   multiple spaces before operator
E222   multiple spaces after operator
E223   tab before operator
E224   tab after operator
E225   operator missing whitespace
E226   arithmetic operator missing whitespace
E227   bitwise or shift operator missing whitespace
E228   modulo missing whitespace
E231   missing whitespace after ','
E241   multiple spaces after ','
E242   tab after ','
E251   unexpected spaces around keyword / parameter equals
E261   at least two spaces before inline comment
E262   inline comment should start with '# '
E265   block comment should start with '# '
E266   too many leading '#' for block comment
E271   multiple spaces after keyword
E272   multiple spaces before keyword
E273   tab after keyword
E274   tab before keyword
E301   expected 1 blank line, found 0
E302   expected 2 blank lines, found 0
E303   too many blank lines (3)
E304   blank lines found after function decorator
E401   multiple imports on one line
E501   line too long (82 > 79 characters)
E502   backslash is redundant between brackets
E701   multiple statements on one line (colon)
E702   multiple statements on one line (semicolon)
E703   statement ends with a semicolon
E704   multiple statements on one line (def)
E711   comparison to None; should be 'if cond is None:'
E712   comparison to True; should be 'if cond is True:' or 'if cond:'
E713   test for membership; should be 'not in'
E714   test for object identity; should be 'is not'
E721   do not compare types, use 'isinstance()'
E731   do not assign a lambda expression, use a def
E901   SyntaxError or IndentationError
E902   IOError

pep8 (warnings)

W191   indentation contains tabs
W291   trailing whitespace
W292   no newline at end of file
W293   blank line contains whitespace
W391   blank line at end of file
W601   .has_key() is deprecated, use 'in'
W602   deprecated form of raising exception
W603   '<>' is deprecated, use '!='
W604   backticks are deprecated, use 'repr()'

pyflakes

F401   module imported but unused
F402   import module from line N shadowed by loop variable
F403   'from module import *' used; unable to detect undefined names
F404   future import(s) name after other statements
F811   redefinition of unused name from line N
F812   list comprehension redefines name from line N
F821   undefined name name
F822   undefined name name in __all__
F823   local variable name ... referenced before assignment
F831   duplicate argument name in function definition
F841   local variable name is assigned to but never used

pylint

C0102   Black listed name "%s"
C0103   Invalid %s name "%s"
C0111   Missing %s docstring
C0112   Empty %s docstring
C0121   Missing required attribute "%s"
C0202   Class method %s should have cls as first argument
C0203   Metaclass method %s should have mcs as first argument
C0204   Metaclass class method %s should have %s as first argument
C0301   Line too long (%s/%s)
C0302   Too many lines in module (%s)
C0303   Trailing whitespace
C0304   Final newline missing
C0321   More than one statement on a single line
C0322   Old: Operator not preceded by a space
C0323   Old: Operator not followed by a space
C0324   Old: Comma not followed by a space
C0325   Unnecessary parens after %r keyword
C0326   %s space %s %s %s\n%s
C1001   Old-style class defined

E0001   (syntax error raised for a module; message varies)
E0011   Unrecognized file option %r
E0012   Bad option value %r
E0100   __init__ method is a generator
E0101   Explicit return in __init__
E0102   %s already defined line %s
E0103   %r not properly in loop
E0104   Return outside function
E0105   Yield outside function
E0106   Return with argument inside generator
E0107   Use of the non-existent %s operator
E0108   Duplicate argument name %s in function definition
E0202   An attribute affected in %s line %s hide this method
E0203   Access to member %r before its definition line %s
E0211   Method has no argument
E0213   Method should have "self" as first argument
E0221   Interface resolved to %s is not a class
E0222   Missing method %r from %s interface
E0235   __exit__ must accept 3 arguments: type, value, traceback
E0501   Old: Non ascii characters found but no encoding specified (PEP 263)
E0502   Old: Wrong encoding specified (%s)
E0503   Old: Unknown encoding specified (%s)
E0601   Using variable %r before assignment
E0602   Undefined variable %r
E0603   Undefined variable name %r in __all__
E0604   Invalid object %r in __all__, must contain only strings
E0611   No name %r in module %r
E0701   Bad except clauses order (%s)
E0702   Raising %s while only classes, instances or string are allowed
E0710   Raising a new style class which does not inherit from BaseException
E0711   NotImplemented raised - should raise NotImplementedError
E0712   Catching an exception which does not inherit from BaseException: %s
E1001   Use of __slots__ on an old style class
E1002   Use of super on an old style class
E1003   Bad first argument %r given to super()
E1004   Missing argument to super()
E1101   %s %r has no %r member
E1102   %s is not callable
E1103   %s %r has no %r member (but some types could not be inferred)
E1111   Assigning to function call which does not return
E1120   No value passed for parameter %s in function call
E1121   Too many positional arguments for function call
E1122   Old: Duplicate keyword argument %r in function call
E1123   Passing unexpected keyword argument %r in function call
E1124   Parameter %r passed as both positional and keyword argument
E1125   Old: Missing mandatory keyword argument %r
E1200   Unsupported logging format character %r (%#02x) at index %d
E1201   Logging format string ends in middle of conversion specifier
E1205   Too many arguments for logging format string
E1206   Not enough arguments for logging format string
E1300   Unsupported format character %r

Python Syntax-Checker Codes

(%#02x) at index %d

E1301   Format string ends in middle of conversion specifier

E1302   Mixing named and unnamed conversion specifiers in format string

E1303   Expected mapping for format string, not %s

E1304   Missing key %r in format string dictionary

E1305   Too many arguments for format string

E1306   Not enough arguments for format string

E1310   Suspicious argument in %s.%s call

F0001   (error prevented analysis; message varies)

F0002   %s: %s (message varies)

F0003   ignored builtin module %s

F0004   unexpected inferred value %s

F0010   error while code parsing: %s

F0202   Unable to check methods signature (%s / %s)

F0220   failed to resolve interfaces implemented by %s (%s)

F0321   Old: Format detection error in %r

F0401   Unable to import %s

I0001   Unable to run raw checkers on built-in module %s

I0010   Unable to consider inline option %r

I0011   Locally disabling %s

I0012   Locally enabling %s

I0013   Ignoring entire file

I0014   Used deprecated directive "pylint:disable-all" or "pylint:disable=all"

I0020   Suppressed %s (from line %d)

I0021   Useless suppression of %s

I0022   Deprecated pragma "pylint:disable-msg" or "pylint:enable-msg"

R0201   Method could be a function

R0401   Cyclic import (%s)

R0801   Similar lines in %s files

R0901   Too many ancestors (%s/%s)

R0902   Too many instance attributes (%s/%s)

R0903   Too few public methods (%s/%s)

R0904   Too many public methods (%s/%s)

R0911   Too many return statements (%s/%s)

R0912   Too many branches (%s/%s)

R0913   Too many arguments (%s/%s)

R0914   Too many local variables (%s/%s)

R0915   Too many statements (%s/%s)

R0921   Abstract class not referenced

R0922   Abstract class is only referenced %s times

R0923   Interface not implemented

RP0001   Messages by category

RP0002   % errors / warnings by module

RP0003   Messages

RP0004   Global evaluation

RP0101   Statistics by type

RP0401   External dependencies

RP0402   Modules dependencies graph

RP0701   Raw metrics

RP0801   Duplication

W0101   Unreachable code

W0102   Dangerous default value %s as argument

W0104   Statement seems to have no effect

W0105   String statement has no effect

W0106   Expression "%s" is assigned to nothing

W0107   Unnecessary pass statement

W0108   Lambda may not be necessary

W0109   Duplicate key %r in dictionary

W0110   map/filter on lambda could be replaced by comprehension

W0120   Else clause on loop without a break statement

W0121   Use raise ErrorClass(args) instead of raise ErrorClass, args.

W0122   Use of exec

W0141   Used builtin function %r

W0142   Used * or ** magic

W0150   %s statement in finally block may swallow exception

W0199   Assert called on a tuple. Did you mean \'assert x,y\'?

W0201   Attribute %r defined outside __init__

W0211   Static method with %r as first argument

W0212   Access to a protected member %s of a client class

W0221   Arguments number differs from %s method

W0222   Signature differs from %s method

W0223   Method %r is abstract in class %r but is not overridden

W0231   __init__ method from base class %r is not called

W0232   Class has no __init__ method

W0233   __init__ method from a non direct base class %r is called

W0234   iter returns non-iterator

W0301   Unnecessary semicolon

W0311   Bad indentation. Found %s %s, expected %s

W0312   Found indentation with %ss instead of %ss

W0331   Use of the <> operator

W0332   Use of "l" as long integer identifier

W0333   Use of the `` operator

W0401   Wildcard import %s

W0402   Uses of a deprecated module %r

W0403   Relative import %r, should be %r

W0404   Reimport %r (imported line %s)

W0406   Module import itself

W0410   __future__ import is not the first non docstring statement

W0511   (warning notes in code comments; message varies)

W0512   Cannot decode using encoding "%s", unexpected byte at position %d

W0601   Global variable %r undefined at the module level

W0602   Using global for %r but no assignment is done

W0603   Using the global statement

W0604   Using the global statement at the module level

W0611   Unused import %s

W0612   Unused variable %r

W0613   Unused argument %r

W0614   Unused import %s from wildcard import

W0621   Redefining name %r from outer scope (line %s)

W0622   Redefining built-in %r

W0623   Redefining name %r from %s in exception handler

W0631   Using possibly undefined loop variable %r

W0632   Possible unbalanced tuple unpacking with sequence%s: …

W0633   Attempting to unpack a non-sequence %s

W0701   Raising a string exception

W0702   No exception type(s) specified

W0703   Catching too general exception %s

W0704   Except does not do anything

W0710   Exception does not inherit from standard "Exception" class

W0711   Exception to catch is the result of a binary "%s" operation

W0712   Implicit unpacking of exceptions is not supported in Python 3

W1001   Use of "property" on an old style class

W1111   Assigning to function call which only returns None

W1201   Specify string format arguments as logging function parameters

W1300   Format string dictionary key should be a string, not %s

W1301   Unused key %r in format string dictionary

W1401   Anomalous backslash in string: \'%s\'. String constant might be missing an r prefix

W1402   Anomalous Unicode escape in byte string: \'%s\'. String constant might be missing an r or u prefix

W1501   "%s" is not a valid mode for open

Created by Devyn Collier Johnson <DevynCJohnson@Gmail.com> (2015 v3) More cheatsheets at DCJTech.info