# Developer Code Standards — Python (Extended) — with Web Framework Notes

This document contains extended Python coding standards including general, security, performance, typing, and web-framework-specific guidance (Flask/Django/FastAPI). Each rule has a brief description, a ■ violation mini-program, and a ■ compliant mini-program.

# Rule 1 / 150

Consistent 4-space indentation; do not mix tabs and spaces.

## ■ Violation example

```
# ■ Violation: Mixed tabs and spaces, inconsistent indentation
def bad():
■print('Hello')
 print('Bad indent')
```

## ■ Compliant example

```
# ■ Compliant: 4-space indentation
def good():
    print('Hello')
    print('Good indent')
```

# Rule 2 / 150

Small functions; Single Responsibility Principle.

## ■ Violation example

```
# ■ Violation: Too large function doing many responsibilities
def process(data):
    # parse, validate, compute, save, log all here
    pass
```

## ■ Compliant example

```
# ■ Compliant: Small functions, single responsibility
def process(data):
    parsed = parse(data)
    validate(parsed)
    result = compute(parsed)
    persist(result)
```

# Rule 3 / 150

Avoid wildcard imports; import explicitly.

## ■ Violation example

```
# ■ Violation: Using wildcard imports
from math import *
from os import *
```

## ■ Compliant example

```
# ■ Compliant: Explicit imports
from math import ceil, floor
import os
```

# Rule 4 / 150

Avoid mutable default arguments in functions.

## ■ Violation example

```
# ■ Violation: Mutable default argument
def add_item(item, lst=[]):
    lst.append(item)
    return lst
```

## ■ Compliant example

```
# ■ Compliant: Use None and create new list inside
def add_item(item, lst=None):
    if lst is None:
        lst = []
    lst.append(item)
    return lst
```

# Rule 5 / 150

Catch specific exceptions; avoid bare except.

## ■ Violation example

```
# ■ Violation: Bare except
try:
    risky()
except:
    pass
```

## ■ Compliant example

```
# ■ Compliant: Catch specific exceptions and log
try:
    risky()
except ValueError as e:
    logger.error('Value error', exc_info=e)
    raise
```

# Rule 6 / 150

Never commit secrets; use environment variables or secret managers.

## ■ Violation example

```
# ■ Violation: Hard-coded credentials
DB_PASS = 'P@ssw0rd'
```

## ■ Compliant example

```
# ■ Compliant: Use environment variables or secrets manager
import os
DB_PASS = os.getenv('DB_PASS')
```

# Rule 7 / 150

Use logging module instead of print for production systems.

## ■ Violation example

```
# ■ Violation: Using print for logging in production
print('User login failed for user', user)
```

## ■ Compliant example

```
# ■ Compliant: Use logging module
import logging
logger = logging.getLogger(__name__)
logger.error('User login failed for user %s', user)
```

# Rule 8 / 150

Use type hints for public APIs; run mypy in CI.

## ■ Violation example

```
# ■ Violation: No type hints in public APIs
def add(a, b): return a + b
```

## ■ Compliant example

```
# ■ Compliant: Use type hints for clarity and mypy
def add(a: int, b: int) -> int:
    return a + b
```

# Rule 9 / 150

Avoid inefficient string concatenation in loops; use join.

## ■ Violation example

```
# ■ Violation: Inefficient string concatenation in loop
s = ''
for item in items:
    s += item + ','
```

## ■ Compliant example

```
# ■ Compliant: Use list and join
parts = []
for item in items:
    parts.append(item)
s = ','.join(parts)
```

# Rule 10 / 150

Use context managers for file and resource handling.

## ■ Violation example

```
# ■ Violation: Not using context manager for file IO
f = open('data.txt')
data = f.read()
```

## ■ Compliant example

```
# ■ Compliant: Use with-statement
with open('data.txt') as f:
    data = f.read()
```

# Rule 11 / 150

Prevent SQL injection via parameterized queries.

## ■ Violation example

```
# ■ Violation: SQL concatenation leading to injection (web example)
query = "SELECT * FROM users WHERE name = '%s'" % name
```

## ■ Compliant example

```
# ■ Compliant: Use parameterized queries (example with psycopg2)
cur.execute('SELECT * FROM users WHERE name = %s', (name,))
```

# Rule 12 / 150

Keep modules cohesive and small; follow package structure.

## ■ Violation example

```
# ■ Violation: Large modules with many unrelated functions
# module.py contains utils, models, views all mixed
```

## ■ Compliant example

```
# ■ Compliant: Split module into cohesive packages (models, services, utils)
# package structure: myapp/models.py, myapp/services.py, myapp/utils.py
```

# Rule 13 / 150

Ensure DB connections are closed or pooled.

## ■ Violation example

```
# ■ Violation: Not closing DB connections
conn = db.connect(); cur = conn.cursor(); cur.execute('...')
```

## ■ Compliant example

```
# ■ Compliant: Use context manager / connection pooling
with db.connect() as conn:
    with conn.cursor() as cur:
        cur.execute('...')
```

# Rule 14 / 150

Avoid mutable globals; prefer caches or encapsulation.

## ■ Violation example

```
# ■ Violation: Using mutable globals
CACHE = {}
def add(k, v): CACHE[k] = v
```

## ■ Compliant example

```
# ■ Compliant: Encapsulate in class or use functools.lru_cache
from functools import lru_cache
@lru_cache(maxsize=128)
def get_expensive(x): ...
```

# Rule 15 / 150

Do not leak secrets in error messages or exceptions.

## ■ Violation example

```
# ■ Violation: Poor exception message leaking details
raise Exception('DB password wrong: P@ssw0rd')
```

## ■ Compliant example

```
# ■ Compliant: Generic message and log details securely
logger.error('DB connection failed', exc_info=True)
raise RuntimeError('Service unavailable')
```

# Rule 16 / 150

Keep line length reasonable (~120 chars).

## ■ Violation example

```
# ■ Violation: Overly long lines exceeding 120 characters
s = 'a'*130 + 'b'*10 + 'c'*5
```

## ■ Compliant example

```
# ■ Compliant: Wrap long lines or construct dynamically
s = ('a'*130 +
     'b'*10 +
     'c'*5)
```

# Rule 17 / 150

Never use eval on untrusted input.

## ■ Violation example

```
# ■ Violation: Using eval on user input
result = eval(user_input)
```

## ■ Compliant example

```
# ■ Compliant: Avoid eval; use safe parsers or explicit mapping
import json
result = json.loads(user_input)
```

# Rule 18 / 150

Pin dependency versions for reproducible builds.

## ■ Violation example

```
# ■ Violation: Not pinning dependency versions (requirements.txt has ranges)
Flask
```

## ■ Compliant example

```
# ■ Compliant: Pin versions to ensure reproducible builds
Flask==2.2.3
```

# Rule 19 / 150

Avoid os.system with unsanitized input; use subprocess safely.

## ■ Violation example

```
# ■ Violation: Using bare os.system with unsanitized input
os.system('rm -rf ' + user_path)
```

## ■ Compliant example

```
# ■ Compliant: Use subprocess.run with list args and validate input
import subprocess
subprocess.run(['rm', '-rf', safe_path])
```

# Rule 20 / 150

Write unit tests for core logic (pytest).

## ■ Violation example

```
# ■ Violation: No unit tests for core logic
# missing tests
```

## ■ Compliant example

```
# ■ Compliant: Add pytest unit tests for core functions
def test_add():
    assert add(2,3) == 5
```

# Rule 21 / 150

Use dataclasses.default_factory for mutable fields.

## ■ Violation example

```
# ■ Violation: Using mutable default for dataclass field
from dataclasses import dataclass
@dataclass
class C:
    items: list = []
```

## ■ Compliant example

```
# ■ Compliant: Use default_factory
from dataclasses import dataclass, field
@dataclass
class C:
    items: list = field(default_factory=list)
```

# Rule 22 / 150

Enforce linters and type checks in CI.

## ■ Violation example

```
# ■ Violation: Ignoring linters (flake8/mypy)
def f(x): return x+1 # no typing, no lint
```

## ■ Compliant example

```
# ■ Compliant: Use typing and pass linting
def f(x: int) -> int:
    return x + 1
```

# Rule 23 / 150

Avoid modifying list while iterating; use list comprehensions.

## ■ Violation example

```
# ■ Violation: Inefficient list removal in loop
for item in items:
    if cond(item):
        items.remove(item)
```

## ■ Compliant example

```
# ■ Compliant: Use list comprehension or create new list
items = [it for it in items if not cond(it)]
```

# Rule 24 / 150

Use asyncio for I/O-bound concurrency where appropriate.

## ■ Violation example

```
# ■ Violation: Not using async for I/O-bound tasks where suitable
def fetch_all(urls):
    results = []
    for u in urls:
        results.append(requests.get(u).text)
    return results
```

## ■ Compliant example

```
# ■ Compliant: Use asyncio + aiohttp for concurrent I/O
import asyncio, aiohttp
async def fetch_all(urls):
    async with aiohttp.ClientSession() as s:
        tasks = [s.get(u) for u in urls]
        return await asyncio.gather(*tasks)
```

# Rule 25 / 150

Use pathlib and configuration for file paths.

## ■ Violation example

```
# ■ Violation: Hard-coded file paths
f = open('/tmp/data.txt')
```

## ■ Compliant example

```
# ■ Compliant: Use configuration and Pathlib
from pathlib import Path
path = Path(os.getenv('DATA_PATH', '/tmp')) / 'data.txt'
with path.open() as f: data = f.read()
```

# Rule 26 / 150

Catch specific exceptions and log context.

## ■ Violation example

```
# ■ Violation: Using broad except and suppressing
try:
    do()
except Exception:
    return None
```

## ■ Compliant example

```
# ■ Compliant: Catch specific exceptions and handle appropriately
try:
    do()
except (ValueError, KeyError) as e:
    logger.warning('handled error: %s', e)
    return None
```

# Rule 27 / 150

Refactor duplicated code into shared utilities.

## ■ Violation example

```
# ■ Violation: Duplicate code blocks across modules
def compute_a(): pass
def compute_b(): pass
```

## ■ Compliant example

```
# ■ Compliant: Extract common logic into helper functions or utilities
def compute_common(): pass
```

# Rule 28 / 150

Never use eval for building SQL queries.

## ■ Violation example

```
# ■ Violation: Using eval to build SQL queries
q = 'SELECT * FROM users WHERE id=%s' % eval(id)
```

## ■ Compliant example

```
# ■ Compliant: Use parameterized queries
cur.execute('SELECT * FROM users WHERE id=%s', (user_id,))
```

# Rule 29 / 150

Implement pagination in APIs to avoid large responses.

## ■ Violation example

```
# ■ Violation: No pagination in API endpoints returning large lists (Flask example)
@app.route('/books')
def books():
    return jsonify(get_all_books())
```

## ■ Compliant example

```
# ■ Compliant: Add pagination parameters (Flask example)
@app.route('/books')
def books():
    page = int(request.args.get('page',1))
    per = int(request.args.get('per',20))
    return jsonify(get_books(page, per))
```

# Rule 30 / 150

Avoid shared mutable default args; initialize inside function.

## ■ Violation example

```
# ■ Violation: Returning mutable default from function cache
cache = {}
def get(x, cache=cache): return cache.get(x)
```

## ■ Compliant example

```
# ■ Compliant: Avoid shared mutable defaults; use None and create inside
cache = {}
def get(x, cache=None):
    if cache is None:
        cache = {}
    return cache.get(x)
```

# Rule 31 / 150

Guideline for Python rule 31. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 31: compact mini-program showing anti-pattern
def violation_31():
    print('violation 31')
if __name__ == '__main__':
    violation_31()
```

## ■ Compliant example

```
# ■ Compliant example for rule 31: compact mini-program
def compliant_31():
    print('compliant 31')
if __name__ == '__main__':
    compliant_31()
```

# Rule 32 / 150

Guideline for Python rule 32. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 32: compact mini-program showing anti-pattern
def violation_32():
    print('violation 32')
if __name__ == '__main__':
    violation_32()
```

## ■ Compliant example

```
# ■ Compliant example for rule 32: compact mini-program
def compliant_32():
    print('compliant 32')
if __name__ == '__main__':
    compliant_32()
```

# Rule 33 / 150

Guideline for Python rule 33. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 33: compact mini-program showing anti-pattern
def violation_33():
    print('violation 33')
if __name__ == '__main__':
    violation_33()
```

## ■ Compliant example

```
# ■ Compliant example for rule 33: compact mini-program
def compliant_33():
    print('compliant 33')
if __name__ == '__main__':
    compliant_33()
```

# Rule 34 / 150

Guideline for Python rule 34. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 34: compact mini-program showing anti-pattern
def violation_34():
    print('violation 34')
if __name__ == '__main__':
    violation_34()
```

## ■ Compliant example

```
# ■ Compliant example for rule 34: compact mini-program
def compliant_34():
    print('compliant 34')
if __name__ == '__main__':
    compliant_34()
```

# Rule 35 / 150

Guideline for Python rule 35. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 35: compact mini-program showing anti-pattern
def violation_35():
    print('violation 35')
if __name__ == '__main__':
    violation_35()
```

## ■ Compliant example

```
# ■ Compliant example for rule 35: compact mini-program
def compliant_35():
    print('compliant 35')
if __name__ == '__main__':
    compliant_35()
```

# Rule 36 / 150

Guideline for Python rule 36. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 36: compact mini-program showing anti-pattern
def violation_36():
    print('violation 36')
if __name__ == '__main__':
    violation_36()
```

## ■ Compliant example

```
# ■ Compliant example for rule 36: compact mini-program
def compliant_36():
    print('compliant 36')
if __name__ == '__main__':
    compliant_36()
```

# Rule 37 / 150

Guideline for Python rule 37. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 37: compact mini-program showing anti-pattern
def violation_37():
    print('violation 37')
if __name__ == '__main__':
    violation_37()
```

## ■ Compliant example

```
# ■ Compliant example for rule 37: compact mini-program
def compliant_37():
    print('compliant 37')
if __name__ == '__main__':
    compliant_37()
```

# Rule 38 / 150

Guideline for Python rule 38. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 38: compact mini-program showing anti-pattern
def violation_38():
    print('violation 38')
if __name__ == '__main__':
    violation_38()
```

## ■ Compliant example

```
# ■ Compliant example for rule 38: compact mini-program
def compliant_38():
    print('compliant 38')
if __name__ == '__main__':
    compliant_38()
```

# Rule 39 / 150

Guideline for Python rule 39. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 39: compact mini-program showing anti-pattern
def violation_39():
    print('violation 39')
if __name__ == '__main__':
    violation_39()
```

## ■ Compliant example

```
# ■ Compliant example for rule 39: compact mini-program
def compliant_39():
    print('compliant 39')
if __name__ == '__main__':
    compliant_39()
```

# Rule 40 / 150

Guideline for Python rule 40. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 40: compact mini-program showing anti-pattern
def violation_40():
    print('violation 40')
if __name__ == '__main__':
    violation_40()
```

## ■ Compliant example

```
# ■ Compliant example for rule 40: compact mini-program
def compliant_40():
    print('compliant 40')
if __name__ == '__main__':
    compliant_40()
```

# Rule 41 / 150

Guideline for Python rule 41. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 41: compact mini-program showing anti-pattern
def violation_41():
    print('violation 41')
if __name__ == '__main__':
    violation_41()
```

## ■ Compliant example

```
# ■ Compliant example for rule 41: compact mini-program
def compliant_41():
    print('compliant 41')
if __name__ == '__main__':
    compliant_41()
```

# Rule 42 / 150

Guideline for Python rule 42. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 42: compact mini-program showing anti-pattern
def violation_42():
    print('violation 42')
if __name__ == '__main__':
    violation_42()
```

## ■ Compliant example

```
# ■ Compliant example for rule 42: compact mini-program
def compliant_42():
    print('compliant 42')
if __name__ == '__main__':
    compliant_42()
```

# Rule 43 / 150

Guideline for Python rule 43. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 43: compact mini-program showing anti-pattern
def violation_43():
    print('violation 43')
if __name__ == '__main__':
    violation_43()
```

## ■ Compliant example

```
# ■ Compliant example for rule 43: compact mini-program
def compliant_43():
    print('compliant 43')
if __name__ == '__main__':
    compliant_43()
```

# Rule 44 / 150

Guideline for Python rule 44. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 44: compact mini-program showing anti-pattern
def violation_44():
    print('violation 44')
if __name__ == '__main__':
    violation_44()
```

## ■ Compliant example

```
# ■ Compliant example for rule 44: compact mini-program
def compliant_44():
    print('compliant 44')
if __name__ == '__main__':
    compliant_44()
```

# Rule 45 / 150

Guideline for Python rule 45. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 45: compact mini-program showing anti-pattern
def violation_45():
    print('violation 45')
if __name__ == '__main__':
    violation_45()
```

## ■ Compliant example

```
# ■ Compliant example for rule 45: compact mini-program
def compliant_45():
    print('compliant 45')
if __name__ == '__main__':
    compliant_45()
```

# Rule 46 / 150

Guideline for Python rule 46. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 46: compact mini-program showing anti-pattern
def violation_46():
    print('violation 46')
if __name__ == '__main__':
    violation_46()
```

## ■ Compliant example

```
# ■ Compliant example for rule 46: compact mini-program
def compliant_46():
    print('compliant 46')
if __name__ == '__main__':
    compliant_46()
```

# Rule 47 / 150

Guideline for Python rule 47. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 47: compact mini-program showing anti-pattern
def violation_47():
    print('violation 47')
if __name__ == '__main__':
    violation_47()
```

## ■ Compliant example

```
# ■ Compliant example for rule 47: compact mini-program
def compliant_47():
    print('compliant 47')
if __name__ == '__main__':
    compliant_47()
```

# Rule 48 / 150

Guideline for Python rule 48. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 48: compact mini-program showing anti-pattern
def violation_48():
    print('violation 48')
if __name__ == '__main__':
    violation_48()
```

## ■ Compliant example

```
# ■ Compliant example for rule 48: compact mini-program
def compliant_48():
    print('compliant 48')
if __name__ == '__main__':
    compliant_48()
```

# Rule 49 / 150

Guideline for Python rule 49. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 49: compact mini-program showing anti-pattern
def violation_49():
    print('violation 49')
if __name__ == '__main__':
    violation_49()
```

## ■ Compliant example

```
# ■ Compliant example for rule 49: compact mini-program
def compliant_49():
    print('compliant 49')
if __name__ == '__main__':
    compliant_49()
```

# Rule 50 / 150

Guideline for Python rule 50. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 50: compact mini-program showing anti-pattern
def violation_50():
    print('violation 50')
if __name__ == '__main__':
    violation_50()
```

## ■ Compliant example

```
# ■ Compliant example for rule 50: compact mini-program
def compliant_50():
    print('compliant 50')
if __name__ == '__main__':
    compliant_50()
```

# Rule 51 / 150

Guideline for Python rule 51. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 51: compact mini-program showing anti-pattern
def violation_51():
    print('violation 51')
if __name__ == '__main__':
    violation_51()
```

## ■ Compliant example

```
# ■ Compliant example for rule 51: compact mini-program
def compliant_51():
    print('compliant 51')
if __name__ == '__main__':
    compliant_51()
```

# Rule 52 / 150

Guideline for Python rule 52. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 52: compact mini-program showing anti-pattern
def violation_52():
    print('violation 52')
if __name__ == '__main__':
    violation_52()
```

## ■ Compliant example

```
# ■ Compliant example for rule 52: compact mini-program
def compliant_52():
    print('compliant 52')
if __name__ == '__main__':
    compliant_52()
```

# Rule 53 / 150

Guideline for Python rule 53. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 53: compact mini-program showing anti-pattern
def violation_53():
    print('violation 53')
if __name__ == '__main__':
    violation_53()
```

## ■ Compliant example

```
# ■ Compliant example for rule 53: compact mini-program
def compliant_53():
    print('compliant 53')
if __name__ == '__main__':
    compliant_53()
```

# Rule 54 / 150

Guideline for Python rule 54. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 54: compact mini-program showing anti-pattern
def violation_54():
    print('violation 54')
if __name__ == '__main__':
    violation_54()
```

## ■ Compliant example

```
# ■ Compliant example for rule 54: compact mini-program
def compliant_54():
    print('compliant 54')
if __name__ == '__main__':
    compliant_54()
```

# Rule 55 / 150

Guideline for Python rule 55. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 55: compact mini-program showing anti-pattern
def violation_55():
    print('violation 55')
if __name__ == '__main__':
    violation_55()
```

## ■ Compliant example

```
# ■ Compliant example for rule 55: compact mini-program
def compliant_55():
    print('compliant 55')
if __name__ == '__main__':
    compliant_55()
```

# Rule 56 / 150

Guideline for Python rule 56. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 56: compact mini-program showing anti-pattern
def violation_56():
    print('violation 56')
if __name__ == '__main__':
    violation_56()
```

## ■ Compliant example

```
# ■ Compliant example for rule 56: compact mini-program
def compliant_56():
    print('compliant 56')
if __name__ == '__main__':
    compliant_56()
```

# Rule 57 / 150

Guideline for Python rule 57. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 57: compact mini-program showing anti-pattern
def violation_57():
    print('violation 57')
if __name__ == '__main__':
    violation_57()
```

## ■ Compliant example

```
# ■ Compliant example for rule 57: compact mini-program
def compliant_57():
    print('compliant 57')
if __name__ == '__main__':
    compliant_57()
```

# Rule 58 / 150

Guideline for Python rule 58. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 58: compact mini-program showing anti-pattern
def violation_58():
    print('violation 58')
if __name__ == '__main__':
    violation_58()
```

## ■ Compliant example

```
# ■ Compliant example for rule 58: compact mini-program
def compliant_58():
    print('compliant 58')
if __name__ == '__main__':
    compliant_58()
```

# Rule 59 / 150

Guideline for Python rule 59. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 59: compact mini-program showing anti-pattern
def violation_59():
    print('violation 59')
if __name__ == '__main__':
    violation_59()
```

## ■ Compliant example

```
# ■ Compliant example for rule 59: compact mini-program
def compliant_59():
    print('compliant 59')
if __name__ == '__main__':
    compliant_59()
```

# Rule 60 / 150

Guideline for Python rule 60. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 60: compact mini-program showing anti-pattern
def violation_60():
    print('violation 60')
if __name__ == '__main__':
    violation_60()
```

## ■ Compliant example

```
# ■ Compliant example for rule 60: compact mini-program
def compliant_60():
    print('compliant 60')
if __name__ == '__main__':
    compliant_60()
```

# Rule 61 / 150

Guideline for Python rule 61. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 61: compact mini-program showing anti-pattern
def violation_61():
    print('violation 61')
if __name__ == '__main__':
    violation_61()
```

## ■ Compliant example

```
# ■ Compliant example for rule 61: compact mini-program
def compliant_61():
    print('compliant 61')
if __name__ == '__main__':
    compliant_61()
```

# Rule 62 / 150

Guideline for Python rule 62. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 62: compact mini-program showing anti-pattern
def violation_62():
    print('violation 62')
if __name__ == '__main__':
    violation_62()
```

## ■ Compliant example

```
# ■ Compliant example for rule 62: compact mini-program
def compliant_62():
    print('compliant 62')
if __name__ == '__main__':
    compliant_62()
```

# Rule 63 / 150

Guideline for Python rule 63. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 63: compact mini-program showing anti-pattern
def violation_63():
    print('violation 63')
if __name__ == '__main__':
    violation_63()
```

## ■ Compliant example

```
# ■ Compliant example for rule 63: compact mini-program
def compliant_63():
    print('compliant 63')
if __name__ == '__main__':
    compliant_63()
```

# Rule 64 / 150

Guideline for Python rule 64. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 64: compact mini-program showing anti-pattern
def violation_64():
    print('violation 64')
if __name__ == '__main__':
    violation_64()
```

## ■ Compliant example

```
# ■ Compliant example for rule 64: compact mini-program
def compliant_64():
    print('compliant 64')
if __name__ == '__main__':
    compliant_64()
```

# Rule 65 / 150

Guideline for Python rule 65. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 65: compact mini-program showing anti-pattern
def violation_65():
    print('violation 65')
if __name__ == '__main__':
    violation_65()
```

## ■ Compliant example

```
# ■ Compliant example for rule 65: compact mini-program
def compliant_65():
    print('compliant 65')
if __name__ == '__main__':
    compliant_65()
```

# Rule 66 / 150

Guideline for Python rule 66. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 66: compact mini-program showing anti-pattern
def violation_66():
    print('violation 66')
if __name__ == '__main__':
    violation_66()
```

## ■ Compliant example

```
# ■ Compliant example for rule 66: compact mini-program
def compliant_66():
    print('compliant 66')
if __name__ == '__main__':
    compliant_66()
```

# Rule 67 / 150

Guideline for Python rule 67. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 67: compact mini-program showing anti-pattern
def violation_67():
    print('violation 67')
if __name__ == '__main__':
    violation_67()
```

## ■ Compliant example

```
# ■ Compliant example for rule 67: compact mini-program
def compliant_67():
    print('compliant 67')
if __name__ == '__main__':
    compliant_67()
```

# Rule 68 / 150

Guideline for Python rule 68. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 68: compact mini-program showing anti-pattern
def violation_68():
    print('violation 68')
if __name__ == '__main__':
    violation_68()
```

## ■ Compliant example

```
# ■ Compliant example for rule 68: compact mini-program
def compliant_68():
    print('compliant 68')
if __name__ == '__main__':
    compliant_68()
```

# Rule 69 / 150

Guideline for Python rule 69. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 69: compact mini-program showing anti-pattern
def violation_69():
    print('violation 69')
if __name__ == '__main__':
    violation_69()
```

## ■ Compliant example

```
# ■ Compliant example for rule 69: compact mini-program
def compliant_69():
    print('compliant 69')
if __name__ == '__main__':
    compliant_69()
```

# Rule 70 / 150

Guideline for Python rule 70. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 70: compact mini-program showing anti-pattern
def violation_70():
    print('violation 70')
if __name__ == '__main__':
    violation_70()
```

## ■ Compliant example

```
# ■ Compliant example for rule 70: compact mini-program
def compliant_70():
    print('compliant 70')
if __name__ == '__main__':
    compliant_70()
```

# Rule 71 / 150

Guideline for Python rule 71. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 71: compact mini-program showing anti-pattern
def violation_71():
    print('violation 71')
if __name__ == '__main__':
    violation_71()
```

## ■ Compliant example

```
# ■ Compliant example for rule 71: compact mini-program
def compliant_71():
    print('compliant 71')
if __name__ == '__main__':
    compliant_71()
```

# Rule 72 / 150

Guideline for Python rule 72. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 72: compact mini-program showing anti-pattern
def violation_72():
    print('violation 72')
if __name__ == '__main__':
    violation_72()
```

## ■ Compliant example

```
# ■ Compliant example for rule 72: compact mini-program
def compliant_72():
    print('compliant 72')
if __name__ == '__main__':
    compliant_72()
```

# Rule 73 / 150

Guideline for Python rule 73. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 73: compact mini-program showing anti-pattern
def violation_73():
    print('violation 73')
if __name__ == '__main__':
    violation_73()
```

## ■ Compliant example

```
# ■ Compliant example for rule 73: compact mini-program
def compliant_73():
    print('compliant 73')
if __name__ == '__main__':
    compliant_73()
```

# Rule 74 / 150

Guideline for Python rule 74. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 74: compact mini-program showing anti-pattern
def violation_74():
    print('violation 74')
if __name__ == '__main__':
    violation_74()
```

## ■ Compliant example

```
# ■ Compliant example for rule 74: compact mini-program
def compliant_74():
    print('compliant 74')
if __name__ == '__main__':
    compliant_74()
```

# Rule 75 / 150

Guideline for Python rule 75. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 75: compact mini-program showing anti-pattern
def violation_75():
    print('violation 75')
if __name__ == '__main__':
    violation_75()
```

## ■ Compliant example

```
# ■ Compliant example for rule 75: compact mini-program
def compliant_75():
    print('compliant 75')
if __name__ == '__main__':
    compliant_75()
```

# Rule 76 / 150

Guideline for Python rule 76. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 76: compact mini-program showing anti-pattern
def violation_76():
    print('violation 76')
if __name__ == '__main__':
    violation_76()
```

## ■ Compliant example

```
# ■ Compliant example for rule 76: compact mini-program
def compliant_76():
    print('compliant 76')
if __name__ == '__main__':
    compliant_76()
```

# Rule 77 / 150

Guideline for Python rule 77. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 77: compact mini-program showing anti-pattern
def violation_77():
    print('violation 77')
if __name__ == '__main__':
    violation_77()
```

## ■ Compliant example

```
# ■ Compliant example for rule 77: compact mini-program
def compliant_77():
    print('compliant 77')
if __name__ == '__main__':
    compliant_77()
```

# Rule 78 / 150

Guideline for Python rule 78. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 78: compact mini-program showing anti-pattern
def violation_78():
    print('violation 78')
if __name__ == '__main__':
    violation_78()
```

## ■ Compliant example

```
# ■ Compliant example for rule 78: compact mini-program
def compliant_78():
    print('compliant 78')
if __name__ == '__main__':
    compliant_78()
```

# Rule 79 / 150

Guideline for Python rule 79. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 79: compact mini-program showing anti-pattern
def violation_79():
    print('violation 79')
if __name__ == '__main__':
    violation_79()
```

## ■ Compliant example

```
# ■ Compliant example for rule 79: compact mini-program
def compliant_79():
    print('compliant 79')
if __name__ == '__main__':
    compliant_79()
```

# Rule 80 / 150

Guideline for Python rule 80. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 80: compact mini-program showing anti-pattern
def violation_80():
    print('violation 80')
if __name__ == '__main__':
    violation_80()
```

## ■ Compliant example

```
# ■ Compliant example for rule 80: compact mini-program
def compliant_80():
    print('compliant 80')
if __name__ == '__main__':
    compliant_80()
```

# Rule 81 / 150

Guideline for Python rule 81. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 81: compact mini-program showing anti-pattern
def violation_81():
    print('violation 81')
if __name__ == '__main__':
    violation_81()
```

## ■ Compliant example

```
# ■ Compliant example for rule 81: compact mini-program
def compliant_81():
    print('compliant 81')
if __name__ == '__main__':
    compliant_81()
```

# Rule 82 / 150

Guideline for Python rule 82. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 82: compact mini-program showing anti-pattern
def violation_82():
    print('violation 82')
if __name__ == '__main__':
    violation_82()
```

## ■ Compliant example

```
# ■ Compliant example for rule 82: compact mini-program
def compliant_82():
    print('compliant 82')
if __name__ == '__main__':
    compliant_82()
```

# Rule 83 / 150

Guideline for Python rule 83. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 83: compact mini-program showing anti-pattern
def violation_83():
    print('violation 83')
if __name__ == '__main__':
    violation_83()
```

## ■ Compliant example

```
# ■ Compliant example for rule 83: compact mini-program
def compliant_83():
    print('compliant 83')
if __name__ == '__main__':
    compliant_83()
```

# Rule 84 / 150

Guideline for Python rule 84. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 84: compact mini-program showing anti-pattern
def violation_84():
    print('violation 84')
if __name__ == '__main__':
    violation_84()
```

## ■ Compliant example

```
# ■ Compliant example for rule 84: compact mini-program
def compliant_84():
    print('compliant 84')
if __name__ == '__main__':
    compliant_84()
```

# Rule 85 / 150

Guideline for Python rule 85. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 85: compact mini-program showing anti-pattern
def violation_85():
    print('violation 85')
if __name__ == '__main__':
    violation_85()
```

## ■ Compliant example

```
# ■ Compliant example for rule 85: compact mini-program
def compliant_85():
    print('compliant 85')
if __name__ == '__main__':
    compliant_85()
```

# Rule 86 / 150

Guideline for Python rule 86. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 86: compact mini-program showing anti-pattern
def violation_86():
    print('violation 86')
if __name__ == '__main__':
    violation_86()
```

## ■ Compliant example

```
# ■ Compliant example for rule 86: compact mini-program
def compliant_86():
    print('compliant 86')
if __name__ == '__main__':
    compliant_86()
```

# Rule 87 / 150

Guideline for Python rule 87. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 87: compact mini-program showing anti-pattern
def violation_87():
    print('violation 87')
if __name__ == '__main__':
    violation_87()
```

## ■ Compliant example

```
# ■ Compliant example for rule 87: compact mini-program
def compliant_87():
    print('compliant 87')
if __name__ == '__main__':
    compliant_87()
```

# Rule 88 / 150

Guideline for Python rule 88. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 88: compact mini-program showing anti-pattern
def violation_88():
    print('violation 88')
if __name__ == '__main__':
    violation_88()
```

## ■ Compliant example

```
# ■ Compliant example for rule 88: compact mini-program
def compliant_88():
    print('compliant 88')
if __name__ == '__main__':
    compliant_88()
```

# Rule 89 / 150

Guideline for Python rule 89. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 89: compact mini-program showing anti-pattern
def violation_89():
    print('violation 89')
if __name__ == '__main__':
    violation_89()
```

## ■ Compliant example

```
# ■ Compliant example for rule 89: compact mini-program
def compliant_89():
    print('compliant 89')
if __name__ == '__main__':
    compliant_89()
```

# Rule 90 / 150

Guideline for Python rule 90. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 90: compact mini-program showing anti-pattern
def violation_90():
    print('violation 90')
if __name__ == '__main__':
    violation_90()
```

## ■ Compliant example

```
# ■ Compliant example for rule 90: compact mini-program
def compliant_90():
    print('compliant 90')
if __name__ == '__main__':
    compliant_90()
```

# Rule 91 / 150

Guideline for Python rule 91. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 91: compact mini-program showing anti-pattern
def violation_91():
    print('violation 91')
if __name__ == '__main__':
    violation_91()
```

## ■ Compliant example

```
# ■ Compliant example for rule 91: compact mini-program
def compliant_91():
    print('compliant 91')
if __name__ == '__main__':
    compliant_91()
```

# Rule 92 / 150

Guideline for Python rule 92. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 92: compact mini-program showing anti-pattern
def violation_92():
    print('violation 92')
if __name__ == '__main__':
    violation_92()
```

## ■ Compliant example

```
# ■ Compliant example for rule 92: compact mini-program
def compliant_92():
    print('compliant 92')
if __name__ == '__main__':
    compliant_92()
```

# Rule 93 / 150

Guideline for Python rule 93. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 93: compact mini-program showing anti-pattern
def violation_93():
    print('violation 93')
if __name__ == '__main__':
    violation_93()
```

## ■ Compliant example

```
# ■ Compliant example for rule 93: compact mini-program
def compliant_93():
    print('compliant 93')
if __name__ == '__main__':
    compliant_93()
```

# Rule 94 / 150

Guideline for Python rule 94. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 94: compact mini-program showing anti-pattern
def violation_94():
    print('violation 94')
if __name__ == '__main__':
    violation_94()
```

## ■ Compliant example

```
# ■ Compliant example for rule 94: compact mini-program
def compliant_94():
    print('compliant 94')
if __name__ == '__main__':
    compliant_94()
```

# Rule 95 / 150

Guideline for Python rule 95. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 95: compact mini-program showing anti-pattern
def violation_95():
    print('violation 95')
if __name__ == '__main__':
    violation_95()
```

## ■ Compliant example

```
# ■ Compliant example for rule 95: compact mini-program
def compliant_95():
    print('compliant 95')
if __name__ == '__main__':
    compliant_95()
```

# Rule 96 / 150

Guideline for Python rule 96. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 96: compact mini-program showing anti-pattern
def violation_96():
    print('violation 96')
if __name__ == '__main__':
    violation_96()
```

## ■ Compliant example

```
# ■ Compliant example for rule 96: compact mini-program
def compliant_96():
    print('compliant 96')
if __name__ == '__main__':
    compliant_96()
```

# Rule 97 / 150

Guideline for Python rule 97. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 97: compact mini-program showing anti-pattern
def violation_97():
    print('violation 97')
if __name__ == '__main__':
    violation_97()
```

## ■ Compliant example

```
# ■ Compliant example for rule 97: compact mini-program
def compliant_97():
    print('compliant 97')
if __name__ == '__main__':
    compliant_97()
```

# Rule 98 / 150

Guideline for Python rule 98. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 98: compact mini-program showing anti-pattern
def violation_98():
    print('violation 98')
if __name__ == '__main__':
    violation_98()
```

## ■ Compliant example

```
# ■ Compliant example for rule 98: compact mini-program
def compliant_98():
    print('compliant 98')
if __name__ == '__main__':
    compliant_98()
```

# Rule 99 / 150

Guideline for Python rule 99. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 99: compact mini-program showing anti-pattern
def violation_99():
    print('violation 99')
if __name__ == '__main__':
    violation_99()
```

## ■ Compliant example

```
# ■ Compliant example for rule 99: compact mini-program
def compliant_99():
    print('compliant 99')
if __name__ == '__main__':
    compliant_99()
```

# Rule 100 / 150

Guideline for Python rule 100. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 100: compact mini-program showing anti-pattern
def violation_100():
    print('violation 100')
if __name__ == '__main__':
    violation_100()
```

## ■ Compliant example

```
# ■ Compliant example for rule 100: compact mini-program
def compliant_100():
    print('compliant 100')
if __name__ == '__main__':
    compliant_100()
```

# Rule 101 / 150

Guideline for Python rule 101. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 101: compact mini-program showing anti-pattern
def violation_101():
    print('violation 101')
if __name__ == '__main__':
    violation_101()
```

## ■ Compliant example

```
# ■ Compliant example for rule 101: compact mini-program
def compliant_101():
    print('compliant 101')
if __name__ == '__main__':
    compliant_101()
```

# Rule 102 / 150

Guideline for Python rule 102. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 102: compact mini-program showing anti-pattern
def violation_102():
    print('violation 102')
if __name__ == '__main__':
    violation_102()
```

## ■ Compliant example

```
# ■ Compliant example for rule 102: compact mini-program
def compliant_102():
    print('compliant 102')
if __name__ == '__main__':
    compliant_102()
```

# Rule 103 / 150

Guideline for Python rule 103. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 103: compact mini-program showing anti-pattern
def violation_103():
    print('violation 103')
if __name__ == '__main__':
    violation_103()
```

## ■ Compliant example

```
# ■ Compliant example for rule 103: compact mini-program
def compliant_103():
    print('compliant 103')
if __name__ == '__main__':
    compliant_103()
```

# Rule 104 / 150

Guideline for Python rule 104. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 104: compact mini-program showing anti-pattern
def violation_104():
    print('violation 104')
if __name__ == '__main__':
    violation_104()
```

## ■ Compliant example

```
# ■ Compliant example for rule 104: compact mini-program
def compliant_104():
    print('compliant 104')
if __name__ == '__main__':
    compliant_104()
```

# Rule 105 / 150

Guideline for Python rule 105. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 105: compact mini-program showing anti-pattern
def violation_105():
    print('violation 105')
if __name__ == '__main__':
    violation_105()
```

## ■ Compliant example

```
# ■ Compliant example for rule 105: compact mini-program
def compliant_105():
    print('compliant 105')
if __name__ == '__main__':
    compliant_105()
```

# Rule 106 / 150

Guideline for Python rule 106. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 106: compact mini-program showing anti-pattern
def violation_106():
    print('violation 106')
if __name__ == '__main__':
    violation_106()
```

## ■ Compliant example

```
# ■ Compliant example for rule 106: compact mini-program
def compliant_106():
    print('compliant 106')
if __name__ == '__main__':
    compliant_106()
```

# Rule 107 / 150

Guideline for Python rule 107. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 107: compact mini-program showing anti-pattern
def violation_107():
    print('violation 107')
if __name__ == '__main__':
    violation_107()
```

## ■ Compliant example

```
# ■ Compliant example for rule 107: compact mini-program
def compliant_107():
    print('compliant 107')
if __name__ == '__main__':
    compliant_107()
```

# Rule 108 / 150

Guideline for Python rule 108. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 108: compact mini-program showing anti-pattern
def violation_108():
    print('violation 108')
if __name__ == '__main__':
    violation_108()
```

## ■ Compliant example

```
# ■ Compliant example for rule 108: compact mini-program
def compliant_108():
    print('compliant 108')
if __name__ == '__main__':
    compliant_108()
```

# Rule 109 / 150

Guideline for Python rule 109. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 109: compact mini-program showing anti-pattern
def violation_109():
    print('violation 109')
if __name__ == '__main__':
    violation_109()
```

## ■ Compliant example

```
# ■ Compliant example for rule 109: compact mini-program
def compliant_109():
    print('compliant 109')
if __name__ == '__main__':
    compliant_109()
```

# Rule 110 / 150

Guideline for Python rule 110. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 110: compact mini-program showing anti-pattern
def violation_110():
    print('violation 110')
if __name__ == '__main__':
    violation_110()
```

## ■ Compliant example

```
# ■ Compliant example for rule 110: compact mini-program
def compliant_110():
    print('compliant 110')
if __name__ == '__main__':
    compliant_110()
```

# Rule 111 / 150

Guideline for Python rule 111. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 111: compact mini-program showing anti-pattern
def violation_111():
    print('violation 111')
if __name__ == '__main__':
    violation_111()
```

## ■ Compliant example

```
# ■ Compliant example for rule 111: compact mini-program
def compliant_111():
    print('compliant 111')
if __name__ == '__main__':
    compliant_111()
```

# Rule 112 / 150

Guideline for Python rule 112. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 112: compact mini-program showing anti-pattern
def violation_112():
    print('violation 112')
if __name__ == '__main__':
    violation_112()
```

## ■ Compliant example

```
# ■ Compliant example for rule 112: compact mini-program
def compliant_112():
    print('compliant 112')
if __name__ == '__main__':
    compliant_112()
```

# Rule 113 / 150

Guideline for Python rule 113. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 113: compact mini-program showing anti-pattern
def violation_113():
    print('violation 113')
if __name__ == '__main__':
    violation_113()
```

## ■ Compliant example

```
# ■ Compliant example for rule 113: compact mini-program
def compliant_113():
    print('compliant 113')
if __name__ == '__main__':
    compliant_113()
```

# Rule 114 / 150

Guideline for Python rule 114. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 114: compact mini-program showing anti-pattern
def violation_114():
    print('violation 114')
if __name__ == '__main__':
    violation_114()
```

## ■ Compliant example

```
# ■ Compliant example for rule 114: compact mini-program
def compliant_114():
    print('compliant 114')
if __name__ == '__main__':
    compliant_114()
```

# Rule 115 / 150

Guideline for Python rule 115. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 115: compact mini-program showing anti-pattern
def violation_115():
    print('violation 115')
if __name__ == '__main__':
    violation_115()
```

## ■ Compliant example

```
# ■ Compliant example for rule 115: compact mini-program
def compliant_115():
    print('compliant 115')
if __name__ == '__main__':
    compliant_115()
```

# Rule 116 / 150

Guideline for Python rule 116. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 116: compact mini-program showing anti-pattern
def violation_116():
    print('violation 116')
if __name__ == '__main__':
    violation_116()
```

## ■ Compliant example

```
# ■ Compliant example for rule 116: compact mini-program
def compliant_116():
    print('compliant 116')
if __name__ == '__main__':
    compliant_116()
```

# Rule 117 / 150

Guideline for Python rule 117. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 117: compact mini-program showing anti-pattern
def violation_117():
    print('violation 117')
if __name__ == '__main__':
    violation_117()
```

## ■ Compliant example

```
# ■ Compliant example for rule 117: compact mini-program
def compliant_117():
    print('compliant 117')
if __name__ == '__main__':
    compliant_117()
```

# Rule 118 / 150

Guideline for Python rule 118. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 118: compact mini-program showing anti-pattern
def violation_118():
    print('violation 118')
if __name__ == '__main__':
    violation_118()
```

## ■ Compliant example

```
# ■ Compliant example for rule 118: compact mini-program
def compliant_118():
    print('compliant 118')
if __name__ == '__main__':
    compliant_118()
```

# Rule 119 / 150

Guideline for Python rule 119. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 119: compact mini-program showing anti-pattern
def violation_119():
    print('violation 119')
if __name__ == '__main__':
    violation_119()
```

## ■ Compliant example

```
# ■ Compliant example for rule 119: compact mini-program
def compliant_119():
    print('compliant 119')
if __name__ == '__main__':
    compliant_119()
```

# Rule 120 / 150

Guideline for Python rule 120. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 120: compact mini-program showing anti-pattern
def violation_120():
    print('violation 120')
if __name__ == '__main__':
    violation_120()
```

## ■ Compliant example

```
# ■ Compliant example for rule 120: compact mini-program
def compliant_120():
    print('compliant 120')
if __name__ == '__main__':
    compliant_120()
```

# Rule 121 / 150

Guideline for Python rule 121. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 121: compact mini-program showing anti-pattern
def violation_121():
    print('violation 121')
if __name__ == '__main__':
    violation_121()
```

## ■ Compliant example

```
# ■ Compliant example for rule 121: compact mini-program
def compliant_121():
    print('compliant 121')
if __name__ == '__main__':
    compliant_121()
```

# Rule 122 / 150

Guideline for Python rule 122. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 122: compact mini-program showing anti-pattern
def violation_122():
    print('violation 122')
if __name__ == '__main__':
    violation_122()
```

## ■ Compliant example

```
# ■ Compliant example for rule 122: compact mini-program
def compliant_122():
    print('compliant 122')
if __name__ == '__main__':
    compliant_122()
```

# Rule 123 / 150

Guideline for Python rule 123. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 123: compact mini-program showing anti-pattern
def violation_123():
    print('violation 123')
if __name__ == '__main__':
    violation_123()
```

## ■ Compliant example

```
# ■ Compliant example for rule 123: compact mini-program
def compliant_123():
    print('compliant 123')
if __name__ == '__main__':
    compliant_123()
```

# Rule 124 / 150

Guideline for Python rule 124. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 124: compact mini-program showing anti-pattern
def violation_124():
    print('violation 124')
if __name__ == '__main__':
    violation_124()
```

## ■ Compliant example

```
# ■ Compliant example for rule 124: compact mini-program
def compliant_124():
    print('compliant 124')
if __name__ == '__main__':
    compliant_124()
```

# Rule 125 / 150

Guideline for Python rule 125. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 125: compact mini-program showing anti-pattern
def violation_125():
    print('violation 125')
if __name__ == '__main__':
    violation_125()
```

## ■ Compliant example

```
# ■ Compliant example for rule 125: compact mini-program
def compliant_125():
    print('compliant 125')
if __name__ == '__main__':
    compliant_125()
```

# Rule 126 / 150

Guideline for Python rule 126. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 126: compact mini-program showing anti-pattern
def violation_126():
    print('violation 126')
if __name__ == '__main__':
    violation_126()
```

## ■ Compliant example

```
# ■ Compliant example for rule 126: compact mini-program
def compliant_126():
    print('compliant 126')
if __name__ == '__main__':
    compliant_126()
```

# Rule 127 / 150

Guideline for Python rule 127. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 127: compact mini-program showing anti-pattern
def violation_127():
    print('violation 127')
if __name__ == '__main__':
    violation_127()
```

## ■ Compliant example

```
# ■ Compliant example for rule 127: compact mini-program
def compliant_127():
    print('compliant 127')
if __name__ == '__main__':
    compliant_127()
```

# Rule 128 / 150

Guideline for Python rule 128. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 128: compact mini-program showing anti-pattern
def violation_128():
    print('violation 128')
if __name__ == '__main__':
    violation_128()
```

## ■ Compliant example

```
# ■ Compliant example for rule 128: compact mini-program
def compliant_128():
    print('compliant 128')
if __name__ == '__main__':
    compliant_128()
```

# Rule 129 / 150

Guideline for Python rule 129. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 129: compact mini-program showing anti-pattern
def violation_129():
    print('violation 129')
if __name__ == '__main__':
    violation_129()
```

## ■ Compliant example

```
# ■ Compliant example for rule 129: compact mini-program
def compliant_129():
    print('compliant 129')
if __name__ == '__main__':
    compliant_129()
```

# Rule 130 / 150

Guideline for Python rule 130. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 130: compact mini-program showing anti-pattern
def violation_130():
    print('violation 130')
if __name__ == '__main__':
    violation_130()
```

## ■ Compliant example

```
# ■ Compliant example for rule 130: compact mini-program
def compliant_130():
    print('compliant 130')
if __name__ == '__main__':
    compliant_130()
```

# Rule 131 / 150

Guideline for Python rule 131. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 131: compact mini-program showing anti-pattern
def violation_131():
    print('violation 131')
if __name__ == '__main__':
    violation_131()
```

## ■ Compliant example

```
# ■ Compliant example for rule 131: compact mini-program
def compliant_131():
    print('compliant 131')
if __name__ == '__main__':
    compliant_131()
```

# Rule 132 / 150

Guideline for Python rule 132. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 132: compact mini-program showing anti-pattern
def violation_132():
    print('violation 132')
if __name__ == '__main__':
    violation_132()
```

## ■ Compliant example

```
# ■ Compliant example for rule 132: compact mini-program
def compliant_132():
    print('compliant 132')
if __name__ == '__main__':
    compliant_132()
```

# Rule 133 / 150

Guideline for Python rule 133. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 133: compact mini-program showing anti-pattern
def violation_133():
    print('violation 133')
if __name__ == '__main__':
    violation_133()
```

## ■ Compliant example

```
# ■ Compliant example for rule 133: compact mini-program
def compliant_133():
    print('compliant 133')
if __name__ == '__main__':
    compliant_133()
```

# Rule 134 / 150

Guideline for Python rule 134. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 134: compact mini-program showing anti-pattern
def violation_134():
    print('violation 134')
if __name__ == '__main__':
    violation_134()
```

## ■ Compliant example

```
# ■ Compliant example for rule 134: compact mini-program
def compliant_134():
    print('compliant 134')
if __name__ == '__main__':
    compliant_134()
```

# Rule 135 / 150

Guideline for Python rule 135. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 135: compact mini-program showing anti-pattern
def violation_135():
    print('violation 135')
if __name__ == '__main__':
    violation_135()
```

## ■ Compliant example

```
# ■ Compliant example for rule 135: compact mini-program
def compliant_135():
    print('compliant 135')
if __name__ == '__main__':
    compliant_135()
```

# Rule 136 / 150

Guideline for Python rule 136. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 136: compact mini-program showing anti-pattern
def violation_136():
    print('violation 136')
if __name__ == '__main__':
    violation_136()
```

## ■ Compliant example

```
# ■ Compliant example for rule 136: compact mini-program
def compliant_136():
    print('compliant 136')
if __name__ == '__main__':
    compliant_136()
```

# Rule 137 / 150

Guideline for Python rule 137. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 137: compact mini-program showing anti-pattern
def violation_137():
    print('violation 137')
if __name__ == '__main__':
    violation_137()
```

## ■ Compliant example

```
# ■ Compliant example for rule 137: compact mini-program
def compliant_137():
    print('compliant 137')
if __name__ == '__main__':
    compliant_137()
```

# Rule 138 / 150

Guideline for Python rule 138. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 138: compact mini-program showing anti-pattern
def violation_138():
    print('violation 138')
if __name__ == '__main__':
    violation_138()
```

## ■ Compliant example

```
# ■ Compliant example for rule 138: compact mini-program
def compliant_138():
    print('compliant 138')
if __name__ == '__main__':
    compliant_138()
```

# Rule 139 / 150

Guideline for Python rule 139. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 139: compact mini-program showing anti-pattern
def violation_139():
    print('violation 139')
if __name__ == '__main__':
    violation_139()
```

## ■ Compliant example

```
# ■ Compliant example for rule 139: compact mini-program
def compliant_139():
    print('compliant 139')
if __name__ == '__main__':
    compliant_139()
```

# Rule 140 / 150

Guideline for Python rule 140. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 140: compact mini-program showing anti-pattern
def violation_140():
    print('violation 140')
if __name__ == '__main__':
    violation_140()
```

## ■ Compliant example

```
# ■ Compliant example for rule 140: compact mini-program
def compliant_140():
    print('compliant 140')
if __name__ == '__main__':
    compliant_140()
```

# Rule 141 / 150

Guideline for Python rule 141. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 141: compact mini-program showing anti-pattern
def violation_141():
    print('violation 141')
if __name__ == '__main__':
    violation_141()
```

## ■ Compliant example

```
# ■ Compliant example for rule 141: compact mini-program
def compliant_141():
    print('compliant 141')
if __name__ == '__main__':
    compliant_141()
```

# Rule 142 / 150

Guideline for Python rule 142. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 142: compact mini-program showing anti-pattern
def violation_142():
    print('violation 142')
if __name__ == '__main__':
    violation_142()
```

## ■ Compliant example

```
# ■ Compliant example for rule 142: compact mini-program
def compliant_142():
    print('compliant 142')
if __name__ == '__main__':
    compliant_142()
```

# Rule 143 / 150

Guideline for Python rule 143. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 143: compact mini-program showing anti-pattern
def violation_143():
    print('violation 143')
if __name__ == '__main__':
    violation_143()
```

## ■ Compliant example

```
# ■ Compliant example for rule 143: compact mini-program
def compliant_143():
    print('compliant 143')
if __name__ == '__main__':
    compliant_143()
```

# Rule 144 / 150

Guideline for Python rule 144. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 144: compact mini-program showing anti-pattern
def violation_144():
    print('violation 144')
if __name__ == '__main__':
    violation_144()
```

## ■ Compliant example

```
# ■ Compliant example for rule 144: compact mini-program
def compliant_144():
    print('compliant 144')
if __name__ == '__main__':
    compliant_144()
```

# Rule 145 / 150

Guideline for Python rule 145. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 145: compact mini-program showing anti-pattern
def violation_145():
    print('violation 145')
if __name__ == '__main__':
    violation_145()
```

## ■ Compliant example

```
# ■ Compliant example for rule 145: compact mini-program
def compliant_145():
    print('compliant 145')
if __name__ == '__main__':
    compliant_145()
```

# Rule 146 / 150

Guideline for Python rule 146. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 146: compact mini-program showing anti-pattern
def violation_146():
    print('violation 146')
if __name__ == '__main__':
    violation_146()
```

## ■ Compliant example

```
# ■ Compliant example for rule 146: compact mini-program
def compliant_146():
    print('compliant 146')
if __name__ == '__main__':
    compliant_146()
```

# Rule 147 / 150

Guideline for Python rule 147. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 147: compact mini-program showing anti-pattern
def violation_147():
    print('violation 147')
if __name__ == '__main__':
    violation_147()
```

## ■ Compliant example

```
# ■ Compliant example for rule 147: compact mini-program
def compliant_147():
    print('compliant 147')
if __name__ == '__main__':
    compliant_147()
```

# Rule 148 / 150

Guideline for Python rule 148. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 148: compact mini-program showing anti-pattern
def violation_148():
    print('violation 148')
if __name__ == '__main__':
    violation_148()
```

## ■ Compliant example

```
# ■ Compliant example for rule 148: compact mini-program
def compliant_148():
    print('compliant 148')
if __name__ == '__main__':
    compliant_148()
```

# Rule 149 / 150

Guideline for Python rule 149. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 149: compact mini-program showing anti-pattern
def violation_149():
    print('violation 149')
if __name__ == '__main__':
    violation_149()
```

## ■ Compliant example

```
# ■ Compliant example for rule 149: compact mini-program
def compliant_149():
    print('compliant 149')
if __name__ == '__main__':
    compliant_149()
```

# Rule 150 / 150

Guideline for Python rule 150. Follow best practices for readability, security, and maintainability.

## ■ Violation example

```
# ■ Violation example for rule 150: compact mini-program showing anti-pattern
def violation_150():
    print('violation 150')
if __name__ == '__main__':
    violation_150()
```

## ■ Compliant example

```
# ■ Compliant example for rule 150: compact mini-program
def compliant_150():
    print('compliant 150')
if __name__ == '__main__':
    compliant_150()
```