

## PEP 8 Essentials for Clean, Readable Code

[PEP 8 – Style Guide for Python Code | peps.python.org](https://peps.python.org/pep-0008/)

### 1. Naming Conventions

Element	Format	Example
Functions & variables	snake_case	load_data(), user_id
Classes	PascalCase	DataProcessor
Constants	UPPER_SNAKE_CASE	MAX_ITERATIONS = 100
Private members	_leading_underscore	_helper_func()

Avoid: camelCase, vague names (x, data1), or abbreviations without context.

### 2. Indentation & Line Length

- Use 4 spaces per indentation level (no tabs).
- Keep lines ≤ 88 characters (or 79 for strict compliance).
- Break long lines using parentheses (preferred) or \:

*# Good*

```
result = process_data(  
    input_file,  
    output_format="json",  
    verbose=True  
)
```

### 3. Whitespace Rules

- Around operators: x = a + b
- After commas: items = [1, 2, 3]
- Not inside brackets: func(arg) (not func( arg ))
- Blank lines:
  - ❖ 2 lines between top-level functions/classes
  - ❖ 1 line between methods in a class

### 4. Imports

Group imports in this order (with blank lines between groups):

```

# Standard library
import os
import sys

# Third-party
import numpy as np
import pandas as pd

# Local/application
from .utils import load_config

```

Never use: from module import \*

## 5. Comments & Docstrings

Comments: Explain why, not what.

```

# Normalize to avoid model bias in skewed data
scores = (scores - scores.mean()) / scores.std()

```

Docstrings: Use triple quotes for all public functions:

```

def predict_churn(features):
    """Predict customer churn probability.

    Args:
        features (pd.DataFrame): Input features.

    Returns:
        np.ndarray: Churn probabilities (0–1).
    """

```

## 6. Common Pitfalls to Avoid

- ❖ ❌ if flag == True: → ✅ if flag:
- ❖ ❌ Mutable defaults: def f(lst=[]): → ✅ def f(lst=None):
- ❖ ❌ Ignoring linter warnings → fix them early!