1. Документ, який встановлює рекомендації щодо стилю коду для проектів Python. PEP8 надає рекомендації щодо форматування коду, іменування змінних, розташування прогалин, використання коментарів і тд

2. Одна порожня стрічка між функціями та класами, щоб відокремити їх один від одного, дві використовуються для розділення фрагментів коду

3. На новий рядок

4. За допомогою оператора from ... import

Або імпортування всього модуля os.path і використання префіксу os.path для доступу до функції join.

5. Пробіл ставиться дорівнює юзається як оператор присвоєння

6. Відкрив для себе Америку, бо думав, що в пайтоні комантар пишеться за допомогою //, а виявляється через #

x = 5 # бла бла бла

7. def add(a, b):

"""

Функція, що додає два числа

Параметри:

a (int): Перше число

b (int): Друге число

Повертає:

int: Сума двох чисел

"""

return a + b

8. Загальні імена, такі як x, y, tmp, data і т. д. Ці імена можуть бути недостатньо зрозумілими та неінформативними, особливо в більших програмах.

Однобуквенні імена, такі як a, b, c і т. д. Їх використання може призводити до плутанини та зрозумілості коду, особливо якщо вони використовуються поза контекстом математичних або індексаційних операцій.

Зарезервовані імена з ключовими словами Python, такі як for, if, while, def і т. д. Не можна використовувати ці слова як імена змінних, оскільки вони мають спеціальне значення в мові Python.

Імена, які перекривають вбудовані функції або модулі, такі як list, str, file і т. д. Краще уникати використання таких імен, оскільки це може призвести до непередбачуваної поведінки програми.

Імена, які містять пробіли або спеціальні символи. Python не дозволяє використовувати пробіли або спеціальні символи, крім знака підкреслення (\_), в іменах змінних. Краще уникати використання таких символів у назвах змінних. # Копіпаст

9. Малі літери і нижне підкреслювання

10. Перша велика літера, без підкреслень, наступне слово тоже велика літера

11. Малі літери і нижнє підкреслювання

12. Великі літери і нижне підкреслювання

13. і так і так варіант підходить, просто isinіtance гручкіший, бо можна перевірити наслідкування класів

14. Перший варіант

15. Обидва варіанти робочі, просто другий більш читабільний і підходить під стандарти написання коду