

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІКТА
Кафедра ЗІ



Звіт
до лабораторної роботи №1

з курсу: «Програмування скриптовими мовами» на тему: «Інсталяція інтерпретатора Python та знайомство з інтегрованим середовищем розробки IDLE. Правила PEP 8»

Виконав:

студент групи КБ-305

Семенчук А.А.

Прийняв:

к.т.н., доцент

Совин Я. Р.

Львів 2024

Мета роботи - ознайомитись з послідовністю налаштування інструментів та створення програмного забезпечення мовою Python в середовищі IDLE. Вивчити правила оформлення коду та інструменти для автоматичного аналізу коду на відповідність вимогам і правилам.

Виконання лабораторної роботи

1. Вивести рядок з ПІБ студента та групою у форматі “Петренко Андрій Степанович КБ23” з допомогою консолі Python, у командному рядку IDLE, через запуск файлу в середовищі IDLE та навести відповідні скріншоти, які підтверджують працездатність інстальованих програм.

```
@Velovo123 →/workspaces/python-labs/lab1 (main) $ python -c 'print("Семенчук Анатолій Анатолійович КБ35")'
Семенчук Анатолій Анатолійович КБ35
@Velovo123 →/workspaces/python-labs/lab1 (main) $ python student_info.py
Семенчук Анатолій Анатолійович КБ35
@Velovo123 →/workspaces/python-labs/lab1 (main) $
```

2. Ознайомитися з PEP 8 (<https://peps.python.org/pep-0008/>).

3. Встановити з допомогою pip літери pylint та ruff.

```
@Velovo123 →/workspaces/python-labs/lab1 (main) $ pylint --version
pylint 3.3.1
astroid 3.3.5
Python 3.12.1 (main, Sep 30 2024, 17:05:21) [GCC 9.4.0]
@Velovo123 →/workspaces/python-labs/lab1 (main) $ ruff --version
ruff 0.7.3
```

4. Переглянути файл test_per8.py, самостійно відзначити порушення правил PEP 8 в ньому і вказати їх у виглядів коментарів у відповідних рядках.

test_per8.py:

```
'''Тестовий файл для ознайомлення з правилами форматування і оформлення коду
згідно Python Enhancement Proposals PEP 8 – Style Guide for Python Code.'''
```

```
import string
import os
import math, time # Порушення: імпорти повинні бути в окремих рядках, один
модуль на рядок.

shift = 3
PI= 3.14 # Порушення: відсутній пробіл до і після знака присвоєння `=`.
def PrintTime(x): # Порушення: назви функцій мають бути у форматі snake_case,
тобто `print_time`.
    if x == True: # Порушення: порівняння з `True` потрібно виконувати через `if
x:` або `if not x:`.
        print( time.ctime() ) # Порушення: зайві пробіли всередині дужок `()``.
    return # Порушення: return без значення не потрібен, якщо немає чого
повертати.

def main (): # Порушення: зайвий пробіл перед `()` у визначенні функції.

    Choice_mode = input("would you like to encode or decode?") # Порушення:
назва змінної повинна бути в форматі snake_case.
    word = input("Please enter text")
```

```

LETTERS = string.ascii_letters + string.punctuation + string.digits #
Порушення: великі літери зазвичай використовуються для констант. Це не єдина
загальноприйнята константа, тому краще використовувати малі літери.
encoded = ""
if Choice_mode == "encode":
    for letter in word:
        if letter == " ":
            encoded = encoded + " " # Порушення: для конкатенації рядків
            краще використовувати `+=`.
        else:
            x = LETTERS.index(letter) + shift
            encoded = encoded + LETTERS[x] # Порушення: для конкатенації
            рядків краще використовувати `+=`.
            y = x + 5 # Порушення: відсутній пробіл до і після знака `=`.

if Choice_mode is "decode": # Порушення: для порівняння рядків
    використовуйте `==`, а не `is`.
    for letter in word:
        if letter == " ":
            encoded = encoded + " " # Порушення: для конкатенації рядків
            краще використовувати `+=`.
        else:
            x = LETTERS.index(letter) - shift # Порушення: відсутній пробіл
            до і після оператора `-`.
            encoded = encoded + LETTERS[x] # Порушення: для конкатенації
            рядків краще використовувати `+=`.
            y = x - 5 # Порушення: відсутній пробіл до і після оператора
            `=`.

print(encoded)
print(Word) # Порушення: змінна `Word` не визначена. Можливо, мала бути
`word`.

if __name__ == '__main__': # Порушень немає, цей рядок відповідає PEP 8.
    main()

```

5. Скоригуйте знайдені порушення та збережіть виправлений файл як test_pep8_my.py.

test_pep8_my.py:

"""Тестовий файл для ознайомлення з правилами форматування і оформлення коду.

Згідно Python Enhancement Proposals PEP 8 – Style Guide for Python Code.

"""

```

import string
import time

```

```

SHIFT = 3
PI = 3.14

```

```

def print_time(x):
    """Виводить поточний час, якщо значення x є істинним."""
    if x:
        print(time.ctime())

```

```

def main():
    """Основна функція для кодування або декодування тексту."""
    choice_mode = input("Would you like to encode or decode? ")

```

```

word = input("Please enter text: ")
letters = string.ascii_letters + string.punctuation + string.digits
encoded = ""
if choice_mode == "encode":
    for letter in word:
        if letter == " ":
            encoded += " "
        else:
            x = letters.index(letter) + SHIFT
            encoded += letters[x]

elif choice_mode == "decode":
    for letter in word:
        if letter == " ":
            encoded += " "
        else:
            x = letters.index(letter) - SHIFT
            encoded += letters[x]

print(encoded)
print(word)

if __name__ == '__main__':
    main()

```

6. Перевірте файл test_пер8.py з допомогою pylint та ruff і порівняйте їх результати зі своїми.

test_пер8.py:

```

@velovoi123 → /workspaces/python-labs/lab1 (main) $ pylint test_пер8.py
***** Module test_пер8
test_пер8.py:1:0: C0301: Line too long (152/100) (line-too-long)
test_пер8.py:11:0: C0301: Line too long (101/100) (line-too-long)
test_пер8.py:11:0: W0311: Bad indentation. Found 3 spaces, expected 4 (bad-indentation)
test_пер8.py:12:0: W0311: Bad indentation. Found 6 spaces, expected 8 (bad-indentation)
test_пер8.py:13:0: W0311: Bad indentation. Found 3 spaces, expected 4 (bad-indentation)
test_пер8.py:18:0: C0301: Line too long (125/100) (line-too-long)
test_пер8.py:20:0: C0301: Line too long (219/100) (line-too-long)
test_пер8.py:25:0: C0301: Line too long (105/100) (line-too-long)
test_пер8.py:28:0: C0301: Line too long (112/100) (line-too-long)
test_пер8.py:36:0: C0301: Line too long (103/100) (line-too-long)
test_пер8.py:38:0: C0301: Line too long (105/100) (line-too-long)
test_пер8.py:39:0: C0301: Line too long (110/100) (line-too-long)
test_пер8.py:6:0: C0410: Multiple imports on one line (math, time) (multiple-imports)
test_пер8.py:8:0: C0103: Constant name "shift" doesn't conform to UPPER_CASE naming style (invalid-name)
test_пер8.py:10:0: C0116: Missing function or method docstring (missing-function-docstring)
test_пер8.py:10:0: C0103: Function name "PrintTime" doesn't conform to snake_case naming style (invalid-name)
test_пер8.py:11:6: C0121: Comparison 'x == True' should be 'x is True' if checking for the singleton value True, or 'x' if testing for truthiness (singleton-comparison)
test_пер8.py:10:0: R1711: Useless return at end of function or method (useless-return)
test_пер8.py:15:0: C0116: Missing function or method docstring (missing-function-docstring)
test_пер8.py:18:4: C0103: Variable name "Choice_mode" doesn't conform to snake_case naming style (invalid-name)
test_пер8.py:20:4: C0103: Variable name "LETTERS" doesn't conform to snake_case naming style (invalid-name)
test_пер8.py:33:7: R0123: In 'Choice_mode is 'decode'', use '==' when comparing constant literals not 'is' ('Choice_mode == 'decode') (literal-comparison)
test_пер8.py:43:10: E0602: Undefined variable 'word' (undefined-variable)
test_пер8.py:29:16: W0612: Unused variable 'y' (unused-variable)
test_пер8.py:5:0: W0611: Unused import os (unused-import)
test_пер8.py:6:0: W0611: Unused import math (unused-import)

-----
Your code has been rated at 0.62/10 (previous run: 0.62/10, +0.00)

```

```

@Velovo123 → /workspaces/python-labs/lab1 (main) $ ruff check test_pep8.py
warning: `one-blank-line-before-class` (D203) and `no-blank-line-before-class` (D211) are incompatible. Ignoring `one-blank-line-before-class`.
warning: `multi-line-summary-first-line` (D212) and `multi-line-summary-second-line` (D213) are incompatible. Ignoring `multi-line-summary-second-line`.
test_pep8.py:1:1: D300 [*] Use triple double quotes `"""`
1 | '''Тестовий файл для ознайомлення з правилами форматування і оформлення коду згідно Python Enhancement Proposals PEP 8 – Style Guide for Python Code.'''
  | ~~~~~
0 |
  | = help: Convert to triple double quotes
test_pep8.py:1:80: E501 Line too long (152 > 79)
1 | '''Тестовий файл для ознайомлення з правилами форматування і оформлення коду згідно Python Enhancement Proposals PEP 8 – Style Guide for Python Code.'''
  | ~~~~~
1 |
test_pep8.py:4:1: I001 [*] Import block is un-sorted or un-formatted
4 | / import string
5 | import os
6 | import math, time # Порушення: імпорти повинні бути в окремих рядках, один модуль на рядок.
7 |
8 | shift = 3
  | ^ I001
9 | PI= 3.14 # Порушення: відсутній пробіл до і після знака присвоєння '='.
10 | def PrintTime(x): # Порушення: назви функцій мають бути у форматі snake case, тобто `print time`.

Found 30 errors.
[*] 7 fixable with the `--fix` option (2 hidden fixes can be enabled with the `--unsafe-fixes` option).

```

7. Перевірте файл test_pep8_my.py з допомогою pylint та ruff і оцініть наскільки добре ви провели виправлення.

test_pep8_my.py:

```

@Velovo123 → /workspaces/python-labs/lab1 (main) $ pylint test_pep8_my.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

@Velovo123 → /workspaces/python-labs/lab1 (main) $ ruff check test_pep8_my.py
warning: `one-blank-line-before-class` (D203) and `no-blank-line-before-class` (D211) are incompatible. Ignoring `one-blank-line-before-class`.
warning: `multi-line-summary-first-line` (D212) and `multi-line-summary-second-line` (D213) are incompatible. Ignoring `multi-line-summary-second-line`.
All checks passed!

```

Висновок: на даній лабораторній роботі я ознайомився з послідовністю налаштування інструментів та створення програмного забезпечення мовою Python в середовищі IDLE. Вивчив правила оформлення коду та інструменти для автоматичного аналізу коду на відповідність вимогам і правилам. Перевірів файл test_pep8.py та test_pep8_my.py з допомогою pylint та ruff. У висновку код файлу test_pep8.py за допомогою pylint був оцінений як: Your code has been rated at 2.50/10. Це означає, що він містить багато помилок і порушень правил PEP8. Інструмент Ruff показав, що знайдено 7 помилок (Found 7 errors). Виправлений код файлу test_pep8_my.py за допомогою pylint був оцінений як: Your code has been rated at 7.78/10, що в порівнянні з кодом до виправлення є набагато коректнішим. Паралельно Ruff вказав, що помилок не виявлено (All checks passed!)