Документування програмного коду за стандартом PEP 257

Документування програмного коду за стандартом PEP 257 включає такі основні правила:

# Загальні правила

* Описова назва: Кожен модуль, клас та функція повинні починатися з описової назви, яка чітко вказує на їх призначення.
* class Circle:
* """Клас для роботи з колами."""
* Однорядкові докстрінги: Для простих функцій або методів, які мають очевидне призначення, використовуються однорядкові докстрінги.
* def calculate\_area(radius):
* """Обчислює площу кола."""
* Багаторядкові докстрінги: Для більш складних або важливих функцій, класів та модулів використовуються багаторядкові докстрінги, які починаються з однотипного речення і продовжуються детальним описом.
* def find\_max\_value(numbers):
* """Повертає найбільше значення у списку чисел.
* Параметри:
* numbers (list): Список чисел.
* Повертає:
* int: Найбільше число у списку.
* """

# Докстрінги модулів

* Опис модуля: Перший рядок докстрінга модуля повинен бути коротким описом його призначення.
* """
* Модуль для роботи з геометричними фігурами.
* """
* Детальний опис: У багаторядкових докстрінгах додається детальний опис, який може включати посилання на інші модулі, класи або функції.
* """
* Модуль для роботи з геометричними фігурами.
* Цей модуль містить класи та функції для роботи з різними геометричними фігурами,
* такими як кола, квадрати і трикутники.
* """

# Докстрінги класів

* Опис класу: Перший рядок докстрінга класу повинен бути коротким описом його призначення.
* class Rectangle:
* """Клас для роботи з прямокутниками."""
* Ініціалізація: Докстрінг класу повинен містити інформацію про метод \_\_init\_\_, який використовується для ініціалізації екземплярів класу.
* class Rectangle:
* """Клас для роботи з прямокутниками.
* Атрибути:
* width (int): Ширина прямокутника.
* height (int): Висота прямокутника.
* """
* def \_\_init\_\_(self, width, height):
* self.width = width
* self.height = height

# Докстрінги методів та функцій

* Опис методу/функції: Перший рядок докстрінга методу або функції повинен бути коротким описом його призначення.
* def calculate\_perimeter(self):
* """Обчислює периметр прямокутника."""
* Аргументи та типи повернення: У багаторядкових докстрінгах додається інформація про аргументи, їх типи та типи значень, які повертаються.
* def resize(self, width, height):
* """Змінює розміри прямокутника.
* Параметри:
* width (int): Нова ширина прямокутника.
* height (int): Нова висота прямокутника.
* """

# Форматування

* Перший рядок: Повинен бути описовим і закінчуватися крапкою.
* """
* Цей модуль містить класи та функції для роботи з геометричними фігурами.
* """
* Далі: Якщо докстрінг багаторядковий, другий рядок повинен бути порожнім, а наступні рядки – містити детальний опис.
* """
* Цей модуль міститькласи та функції для роботи з геометричними фігурами.
* Цей модуль містить функції для обчислення площі, периметра та інших властивостей
* різних геометричних фігур.
* """

Інструменти автоматичної генерації документації використанням стандарту PEP 257

pDoc – це один із інструментів автоматичної генерації документації, який використовує стандарт PEP 257 для створення детальних докстрінгів для Python-модулів. Він забезпечує узгодженість стилю опису, що значно полегшує розуміння коду та його супровід. pDoc автоматично додає порожній рядок після першого описового рядка багаторядкового докстрінга, що відповідає вимогам PEP 257. Це дозволяє розробникам зосередитися на написанні коду, а не на формальностях документації.

Для генерації документацій використовувалась наступна команда:

python -m pdoc botnlai5.py -o docs

Що таке JSON?

Опис формату JSON

JSON (JavaScript Object Notation) — це текстовий формат обміну даними, що легко читається людиною та обробляється комп'ютерами. JSON був розроблений для спрощення передачі даних між сервером та клієнтом, а також для зберігання структурованої інформації в текстовому вигляді.

# Структура JSON

JSON складається з пар ключ-значення, де ключі є рядками, а значення можуть бути різних типів, включаючи числа, рядки, логічні значення, масиви та об'єкти. Ось приклад структури JSON:

```json

{

"ім'я": "Іван",

"вік": 30,

"одружений": true,

"діти": ["Анна", "Микола"],

"адреса": {

"місто": "Київ",

"вулиця": "Хрещатик",

"номер будинку": 1

}

}

```

# Основні типи даних JSON

* Рядки: Рядки записуються в подвійні лапки. Наприклад: "Привіт, світ!"
* Числа: Числа можуть бути цілими або дробовими. Наприклад: 42, 3.14
* Логічні значення: Логічні значення можуть бути true або false.
* Масиви: Масиви є впорядкованими списками значень. Наприклад: [1, 2, 3, "чотири"]
* Об'єкти: Об'єкти містять пари ключ-значення. Наприклад: {"ключ1": "значення1", "ключ2": "значення2"}
* Null: Null використовується для позначення відсутності значення.

# Переваги використання JSON

* Простота: JSON має легку та читабельну структуру, що спрощує його використання та розуміння.
* Міжплатформенність: JSON підтримується багатьма мовами програмування, включаючи JavaScript, Python, Java, C# та інші.
* Універсальність: JSON може використовуватися для передачі даних між різними системами та платформами.
* Широке використання: JSON є стандартом де-факто для передачі даних в веб-додатках та API.

# Використання JSON у програмуванні

JSON широко використовується у веб-розробці для передачі даних між сервером та клієнтом. Наприклад, під час виконання AJAX-запитів, сервер може відправити клієнту дані у форматі JSON, які потім будуть оброблені JavaScript-кодом на стороні клієнта. Ось приклад такого запиту:

```javascript

fetch("[URL]")

.then(response => response.json())

.then(data => {

console.log(data);

})

.catch(error => {

console.error("Error:", error);

});

```

# Висновок

JSON — це зручний і простий у використанні формат обміну даними, який широко застосовується в сучасній веб-розробці. Завдяки своїй простоті, універсальності та підтримці багатьма мовами програмування, JSON став стандартом для передачі структурованої інформації між різними системами та платформами.

Для отримання більш детальної інформації про формат JSON, ви можете звернутися до наступного джерела:

JSON. Вікіпедія: вільна енциклопедія. URL: [URL] (дата звернення: 21 квітня 2025 року).