

Логування




Навіщо потрібне логування?

1. **Моніторинг:** Відстеження стану програми в реальному часі.
2. **Налагодження:** Допомогає знаходити та виправляти помилки.
3. **Аналіз:** Збір даних про роботу програми для подальшого аналізу.
4. **Аудит:** Логування дій користувачів чи системних змін для забезпечення безпеки.



Рівні логів

- **DEBUG:** Для налагодження. Детальна інформація про роботу програми.
- **INFO:** Інформаційні повідомлення, що описують нормальний хід роботи.
- **WARNING:** Попередження про потенційні проблеми.
- **ERROR:** Помилки, які не зупиняють роботу програми.
- **CRITICAL:** Критичні помилки, які можуть зупинити програму.



Логери (Logger):

Відповідають за генерацію повідомлень логів.

Обробники (Handler):

Визначають, куди спрямовувати логи (файли, консоль, база даних тощо).

Форматери (Formatter):

Налаштовують формат повідомлень (дата, час, рівень, текст).

logging.basicConfig()

logging.basicConfig() - логи выводятся в консоль

logging.basicConfig(filename='app.log') - логи записываются до файла "app.log"

logging.basicConfig(filename='app.log', filemode='a') - 'w' — перезаписує файл, 'a' — додає логи у кінець файлу.

logging.basicConfig(level=logging.INFO) - мінімальний рівень (DEBUG, INFO, WARNING, ERROR, CRITICAL)



```
logging.basicConfig(format='%(asctime)s - %(levelname)s - %(message)s')
```

datefmt: Формат даты для параметра **%(asctime)s**

```
logging.basicConfig( format='%(asctime)s - %(message)s',  
datefmt='%Y-%m-%d %H:%M:%S' )
```

```
logging.basicConfig(handlers=[ logging.FileHandler('app.log'),  
logging.StreamHandler() ])
```



logging.basicConfig(

```
filename='app.log',           # Ім'я файлу для логів
filemode='w',                 # Перезапис файлу при кожному
запуску level=logging.INFO,   # Мінімальний рівень
format='%(asctime)s - %(levelname)s - %(message)s',
datefmt='%Y-%m-%d %H:%M:%S'   # Формат дати
)
```




```
logging.basicConfig(
```


```
level=logging.DEBUG,
```

Усі рівні логів

```
format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
```

```
datefmt='%H:%M:%S'
```

```
)
```





```
file_handler = logging.FileHandler('app.log')  
console_handler = logging.StreamHandler()
```

```
formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
```

```
file_handler.setFormatter(formatter) console_handler.setFormatter(formatter)
```

```
logger = logging.getLogger()  
logger.setLevel(logging.DEBUG)  
logger.addHandler(file_handler)  
logger.addHandler(console_handler)
```



```
formatter = logging.Formatter('%(asctime)s - %(levelname)s - %  
(message)s - %(name)s - %(filename)s - %(lineno)d - %(funcName)s')
```

1. `%(asctime)s`

Дата та час запису логу.

Формат можна змінити через параметр `datefmt` у `logging.Formatter`.

2. `%(levelname)s`

Рівень логування (наприклад, `DEBUG`, `INFO`, `WARNING`, `ERROR`, `CRITICAL`).

3. `%(message)s`

Основний текст повідомлення, що передається у функції логування (наприклад, `logger.info("Message")`).

4. `%(name)s`

Ім'я логера, який генерує повідомлення.

Якщо ім'я логера не задано, використовується `root`.

5. `%(filename)s`

Назва файлу, з якого було викликано логування (без повного шляху).

6. `%(lineno)d`

Номер рядка в коді, де було викликано логування.

7. `%(funcName)s`

Назва функції, з якої було викликано логування.

