Информация о логировании.

Модуль `logging` в Python предоставляет гибкий механизм логирования для приложений. **Логирование** - это запись информации о ходе выполнения программы, которая может быть полезной при разработке, тестировании и обслуживании.

Вот основные компоненты модуля `logging`:

1. **Logger (Логгер):** Главный объект, который предоставляет интерфейс для отправки логов. В вашем коде вы будете создавать экземпляры `Logger` для каждого модуля или компонента, который требует логирования.
2. **Handler (Обработчик):** Определяет, куда будут направляться сообщения о логах. Например, сообщения могут выводиться в консоль, записываться в файл или отправляться по электронной почте.
3. **Formatter (Форматировщик):** Определяет, в каком виде будет представлена каждая запись лога. Вы можете настроить форматирование, чтобы включить в логи временные метки, уровни логирования и другую информацию.

Пример кода:

```
import logging

# Настроим корневой логгер

logging.basicConfig(level=logging.DEBUG)  # Установка уровня логирования
```

```
# Создадим логгер для текущего модуля

logger = logging.getLogger(__name__)

# если нужен то
# Настроим корневой логгер

logging.basicConfig(filename='example.log',
level=logging.DEBUG)  # Установка уровня логирования и
файла для записи

# Создадим обработчик, который будет записывать
сообщения в файл

file_handler = logging.FileHandler('example.log')

file_handler.setLevel(logging.INFO)  # Устанавливаем
уровень логирования для обработчика


# Создадим обработчик, который будет выводить сообщения
в консоль

console_handler = logging.StreamHandler()

console_handler.setLevel(logging.INFO)  # Устанавливаем
уровень логирования для обработчика

# Создадим форматировщик

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

# Привяжем обработчик к форматировщику
```

```
console_handler.setFormatter(formatter)

# Привяжем обработчик к логгеру

logger.addHandler(console_handler)

# Примеры логирования

logger.debug('Это сообщение уровня DEBUG')

logger.info('Это информационное сообщение')

logger.warning('Это предупреждение')

logger.error('Это сообщение об ошибке')

logger.critical('Это критическая ошибка')
```