

5 модуль. Задания

Глава 5.2. asyncio.run

1. Что делает функция `asyncio.run()` в `asyncio`?

- Запускает параллельный процесс
- Открывает новое окно в браузере
- Создает новый поток выполнения
- Выполняет переданную корутину, управляя циклом событий

Правильный ответ: Выполняет переданную корутину, управляя циклом событий

2. Какие ограничения имеет функция `asyncio.run()`?

- Нельзя использовать внутри не-корутин
- Нельзя использовать внутри циклов
- Нельзя вызывать, когда другой цикл событий работает в том же потоке
- Нельзя использовать внутри классов

Правильный ответ: Нельзя вызывать, когда другой цикл событий работает в том же потоке

3. Что произойдет, если передать обычную функцию в `asyncio.run()`?

- Вы получите ошибку
- Функция будет проигнорирована
- Функция будет преобразована в корутину
- Функция будет выполнена синхронно

Правильный ответ: Вы получите ошибку

Глава 5.3. asyncio.sleep

1. Что делает функция `asyncio.sleep()`?

- Блокирует выполнение всей программы
- Ускоряет выполнение корутины
- Приостанавливает выполнение текущей корутины на указанное время
- Завершает выполнение корутины

Правильный ответ: Приостанавливает выполнение текущей корутины на указанное время

2. Возвращает ли `asyncio.sleep()` значение?

- Время ожидания
- Ошибку
- None
- Текущее время

Правильный ответ: None

Глава 5.4. asyncio.create_task

1. Что делает функция `asyncio.create_task()`?

- Запускает корутину в новом потоке
- Запускает цикл событий
- Создает задачу из корутины
- Создает новый поток

Правильный ответ: Создает задачу из корутины

2. Как отменить задачу, созданную `asyncio.create_task()`?

- `task.stop()`
- `task.cancel()`
- `asyncio.stop()`
- `asyncio.cancel()`

Правильный ответ: `task.cancel()`

Глава 5.5. asyncio.wait_for

1. Что делает функция `asyncio.wait_for()`?
 - Запускает корутину без ожидания
 - Ожидает выполнение без ограничения по времени
 - Отменяет выполнение корутины
 - Ожидает выполнение корутины в течение указанного времени

Правильный ответ: Ожидает выполнение корутины в течение указанного времени

2. Какое исключение выбрасывается при превышении тайм-аута?

- `asyncio.CancelledError`
- `asyncio.LimitOverrunError`
- `asyncio.InvalidStateError`
- `asyncio.TimeoutError`

Правильный ответ: `asyncio.TimeoutError`

Глава 5.6. asyncio.gather

1. Что делает функция `asyncio.gather()`?

- Запускает только корутины
- Запускает только задачи
- Группирует несколько `awaitable` для одновременного выполнения
- Запускает одну корутину за раз

Правильный ответ: Группирует несколько `awaitable` для одновременного выполнения

2. В каком порядке возвращаются результаты `asyncio.gather()`?

- В случайном порядке
- В порядке выполнения
- В порядке передачи аргументов
- В обратном порядке

Правильный ответ: В порядке передачи аргументов

Глава 5.7. asyncio.wait

1. Что возвращает функция `asyncio.wait()`?

- Время выполнения задач
- Кортеж (`done, pending`)
- Количество задач
- Одно множество задач

Правильный ответ: Кортеж (`done, pending`)

2. Что означает `return_when=asyncio.FIRST_COMPLETED`?

- После завершения всех задач
- После первого исключения
- После завершения последней задачи
- После завершения первой задачи

Правильный ответ: После завершения первой задачи