

Spis treści

Wstęp	7
1. Prezentacja Asyncio	11
Restauracja ThreadBotów	11
Epilog	15
Jakie problemy stara się rozwiązywać Asyncio?	16
2. Prawda o wątkach	19
Zalety stosowania wątków	20
Wady stosowania wątków	21
Studium przypadku: roboty i sztuce	24
3. Asyncio — przegląd informacji	31
Szybki start	32
Wieża Asyncio	37
Koprocedury	40
Nowe słowa kluczowe async def	41
Nowe słowo kluczowe await	43
Pętla zdarzeń	46
Klasy Task i Future	48
Kilka słów o terminologii	51
Asynchroniczne menedżery kontekstu: async with	54
Zastosowanie modułu contextlib	55
Iteratory asynchroniczne: async for	58
Prostszy kod dzięki użyciu generatorów asynchronicznych	61
Asynchroniczne wyrażenia listowe	62
Rozpoczynanie i kończenie (łagodne!)	64
Do czego służy argument return_exceptions=True funkcji gather()?	68
Sygnały	70
Oczekiwanie na egzekutor podczas procesu kończenia	74
4. 20 bibliotek Asyncio, których nie używasz (ale... mniejsza z tym)	81
Strumienie (biblioteka standardowa)	82
Studium przypadku: kolejka komunikatów	82
Studium przypadku: poprawa kolejki komunikatów	89
Framework Twisted	93
Kolejka Janus	95

aiohhttp	97
Studium przypadku: „Witaj, świecie!”	97
Studium przypadku: mechanizm zbierania doniesień	97
0MQ (ZeroMQ)	101
Studium przypadku: obsługa wielu gniazd	102
Studium przypadku: monitorowanie wydajności aplikacji	106
asynpcg i Sanic	113
Studium przypadku: unieważnienie pamięci podręcznej	117
Inne biblioteki i zasoby	127
5. Przemyślenia końcowe	129
A. Krótka historia programowania asynchronicznego w języku Python	131
Na początku było asyncore	131
Ścieżka do rodzimych koprocedur	133
B. Materiały uzupełniające	135
Przykład ze sztuczkami z wykorzystaniem asyncio	135
Materiały dodatkowe do przykładu z mechanizmem zbierania doniesień	137
Materiały uzupełniające studium przypadku z ZeroMQ	138
Obsługa wyzwalaczy bazy danych na potrzeby studium przypadku użycia bibliotek asynpcg	140
Materiał uzupełniający do przykładu z frameworkiem Sanic: aelapsed i aprofiler	142

oprac. BPK