

Szkolenie: Python Academy Optimizing Python Programs



Cel szkolenia:

Python jest interpretowanym językiem programowania - kod źródłowy jest tłumaczony w przenośny bajt-kod. Ten pomysł w połączeniu z innymi właściwościami Pythona stawia go na prowadzeniu w porównaniu z innymi językami. Jednak w rezultacie szybkość wykonania może być odczuwane mniejsza dla niektórych typów aplikacji, w porównaniu z językami kompilowanymi. Optymalizacja często może znacząco podnieść wydajność oprogramowania w Pythonie.

Plan szkolenia:

- Wprowadzenie do optymalizacji.
 - Strategie optymalizacyjne - Benchmarking z Pystone
 - Profilowanie CPU z cProfile
 - Pomiar zużycia pamięci z Guppy_PE Framework
- Algorytmy i anty-wzorce projektowe
 - Przykłady algorytmów, które działają wyjątkowo wolno lub szybko w Pythonie.
- Odpowiednie struktury danych
 - Porównanie wbudowanych struktur danych: list, zbiorów, podwójnie zakończonych kolejek, standardowych słowników.
 - Notacja dużego O w przykładach.
- Caching
 - Deterministyczne i niedeterministyczne spojrzenie na caching.
 - Proces tworzenia dekoratorów.
- Przykład - znaleźliśmy numerycznie i obliczeniowo wymagający problem i zaimplementujemy go w czystym Pythonie. Później popatrzymy na możliwości algorytmicznej poprawy szybkości obliczeń.
- Testowanie prędkości - znajdowanie rozwiązań do prawidłowego pomiaru czasu wykonania programu.
- Pyco - 'just-in-time-compiler' (JIT), pozwalający na tłumaczenie części kodu bajtowego na kod maszynowy. Przykłady celem pokazania możliwości użycia Pyco.
- Obliczenia numeryczne z NumPy - podstawy użycia biblioteki.
- Używanie wielu jednostek obliczeniowych z PyProcessing/Multiprocessing.
- Kombinacja strategii optymalizacyjnych

- Przegląd rozszerzeń do Pythona z innych języków.

Wymagania:

- Podstawowa znajomość języka Python.

Poziom trudności



Certyfikaty:

Uczestnicy otrzymują po zakończeniu szkolenia zaświadczenie o ukończeniu autoryzowanego kursu Python Academy

Prowadzący:

Autoryzowany wykładowca Python Academy.