

Wprowadzenie do Pythona - część 1

(Uwaga – kolejność przykładów w pliku Wprowadzenie.py, ze względów przejrzystości, różni się minimalnie od kolejności opisu w poniższym tekście.)

Ogólne informacje

Skąd pobrać Pythona?

Najlepiej ze strony twórców: <http://www.python.org/download/>

Znajdują się tam wersje dla większości popularnych platform. Warto też zwrócić uwagę, że systemy takie jak Ubuntu i Mac OS X mają Pythona wbudowanego już po instalacji.

Jakiego programu używać do pisania w Pythonie?

Najlepiej edytora, który jest w stanie kolorować składnię i wspomagać nas w pisaniu, czyli potrafi podpowiadać co wpisać, ułatwia wstawianie tabulatorów.

Osobiście używam następujących programów:

- Notepad++ (pod systemem Microsoft Windows);
- Geany (na Ubuntu);
- TextMate (na Mac OS X);

Oczywiście wybór, zależy od osobistych preferencji.

Jak uruchomić program napisany w Pythonie?

Pliki Pythona mają rozszerzenie „.py”. Uruchamiamy je komendą „python nazwa_pliku.py”. W systemie Windows wystarczy dwukrotnie kliknąć ikonę pliku, aby się on uruchomił.

Mała uwaga: W systemie Windows okno programu zamknie się tuż po jego zakończeniu, dlatego warto dodać na jego koniec, np. oczekiwanie na wpisanie czegokolwiek z klawiatury, dzięki temu, będziemy mogli spokojnie obejrzeć końcowy efekt pracy programu.

Jakie są podstawowe zasady pisania w Pythonie?

„Znaczki w kodzie”:

- # - oznacza komentarz na jedną linię;
- """ ... """ - komentarz na wiele linii;
- = - operator przypisania wartości (UWAGA! Nie prównania!);
- -, +, *, / - w ogólności działają jak operatory matematyczne;
- . - dostęp do właściwości/metod klasy; (w dużym uproszczeniu);

Piotr Janus

piotr.nex@gmail.com

Porównywanie:

- < - mniejsze od;
- > - większe od;
- <= - mniejsze lub większe od;
- >= - większe lub mniejsze od;
- == - równe;
- != (lub <>) - różne;
- is - identyczne obiekty;
- is not - różne obiekty;

Uwagi dodatkowe

W Pythonie mają znaczenie białe znaki (spacje, tabulatory, itp.). Pisząc program należy pamiętać o prawidłowych wcięciach w kodzie, często problem nie leży w kodzie, a właśnie we wcięciach.

Na początku pliku, powinna znaleźć się informacja o kodowaniu znaków, zapobiega to wielu dziwnym problemom w trakcie wykonywania programu. Osobiście polecam „#coding=UTF-8”, ponieważ obsługuje praktycznie wszystkie znaki ze wszystkich języków.

Zmienne

Zmienna to „byt” w programie posiadający nazwę i wartość (w Pythonie ograniczymy się do tego).

Nazwa jest niezmienna po jej ustaleniu. UWAGA! Nazwa MUSI zaczynać się od litery alfabetu łacińskiego. Może też zawierać cyfry i znaki podkreślenia „_”. Nie może zawierać innych znaków.

Wartość zmiennej posiada swój typ, dzięki któremu program jest w stanie odróżniać dane na których pracuje. (Może być to liczba, tekst, lub jakaś skomplikowana struktura) W Pythonie typy zmiennych są dynamiczne, jednak czasami konieczna jest ich ręczna konwersja, np. po wczytaniu liczby z klawiatury, Python uważa, że ta liczba to tekst (co zdroworozsądkowo jest prawdą).

Wartość zmiennej możemy w dowolnym momencie odczytać lub zmienić. Możemy jednej zmiennej przypisać wartość innej zmiennej. Przy przypisaniu można korzystać z wyrażeń arytmetycznych oraz procedur (do wyjaśnienia później).

Listy

Listy to specjalne obiekty, które pozwalają na przechowywanie ciągów (list) innych elementów.

Deklaruje się je podobnie jak zmienne.

Nie posiadają one limitów długości, ani nie narzucają jakiego typu dane możemy w nich umieszczać.

UWAGA! Przy dodawaniu zmiennej do listy, dodawana jest wartość tej zmiennej a nie sama zmienna. (Czyli zmiana wartości zmiennej po jej dodaniu do listy, nie powoduje zmiany wartości na liście.)

Piotr Janus

piotr.nex@gmail.com

Instrukcja warunkowa

Warunek `if` **WARUNEK** `else` pozwala na wykonanie pewnych czynności w zależności od tego czy jakiś fakt ma miejsce. Blok `if` jest wykonany kiedy warunek jest prawdziwy, a `else` w przeciwnym wypadku.

UWAGA! Można używać `if` bez `else`. (Wtedy jeżeli warunek nie jest spełniony, program wykonuje się dalej.)

Pętle

Pętla for

Pętla `for` **ELEMENT** `in` **ELEMENTS** pozwala na manipulację każdym po kolei elementem ze wszystkich elementów dostępnych w danej strukturze (w przykładach były to listy).

Pętla while

Pętla `while` **WARUNEK** wykonuje się dopóki warunek jest spełniony (w szczególności: pętla `while True`, wykonywać się będzie w nieskończoność).