

Przygoda z Pythonem

Zapraszamy do pierwszego wydania naszego magazynu poświęconego Pythonowi – potężnemu i uniwersalnemu językowi programowania.

Według indeksu TIOBE niedawno Python został najpopularniejszym językiem programowania na świecie, wyprzedzając takich konkurentów jak C, C++ czy Java, przy czym jego popularność stale rośnie. Nietrudno zrozumieć, dlaczego tak się dzieje. Dzięki prostej składni i względnie niewielu elementom języka, można go stosunkowo szybko opanować w stopniu podstawowym, a następnie stopniowo zwiększać swoje umiejętności.

Drugim ważnym czynnikiem jest obecność wielu przydatnych struktur danych oraz powiązanych funkcji do pracy z nimi, dzięki którym czynności, które w innych językach wymagają napisania wielu wierszy kodu, w Pythonie można wykonać jedną liniijką. Dodatkowo w językach, w których ręcznie zarządzamy pamięcią, można popełnić wiele poważnych w skutkach błędów. Wystarczy porównać np. przetwarzanie danych tekstowych w Pythonie i w C, C++ czy nawet w Javie, by zrozumieć, dlaczego wielu programistów, mając wybór, decyduje się na tę pierwszą opcję.

Do sukcesu Pythona niewątpliwie przyczyniła się bogata biblioteka standardowa (reklamowana hasłem „baterie w zestawie”). Wiele czynności, które gdzie indziej wymagałyby instalacji dodatkowego oprogramowania czy nawet napisania własnego, można w Pythonie wykonać od razu. Co więcej, ogromna większość funkcji znajdujących się w bibliotece standardowej działa na wszystkich obsługiwanych platformach. Oznacza to, że program, który napiszę w Windows, można bez specjalnych modyfikacji uruchomić na macOS-ie, Linuksie czy BSD, uzyskując te same lub bardzo podobne rezultaty.

Python działa wszędzie. Znajdziemy go nie tylko w Internecie, we frameworkach takich jak Flask czy Django, ale również na niewielkich płytkach: w Pythonie możemy programować LEGO Mindstorms/Spike, BBC:microw, a także wiele różnych

mikrokontrolerów obsługujących MicroPythona. Innymi słowy, jeśli chcemy zbudować własną stację pogodową czy zdalnie sterowanego robota, nie musimy uczyć się nowego języka programowania – możemy wykorzystać znajomość Pythona.

Pierwsze wydanie naszego magazynu poświęconego Pythonowi składa się dwóch zasadniczych części. Pierwsza dotyczy podstaw języka, począwszy od instalacji, wyboru edytora czy środowiska programistycznego, poprzez naukę typów i struktur danych oraz mechanizmów kontroli przepływu programu. Część druga zawiera kilka bardzo różnych przykładów zastosowań Pythona, od multimediiów po Internet.

Całość jest przeznaczona do czytania sekwencyjnego: nowe koncepcje wprowadzane są stopniowo i wymagają przyswojenia sobie przedstawionej poprzednio wiedzy. Najwygodniej jest rozłożyć magazyn obok komputera i samodzielnie testować przykładowy kod. Nie powinno być to kłopotliwe: dołożyliśmy wszelkich starań, by był on możliwie zwięzły i wszystkie przykłady mieściły się na jednym ekranie.

Oczywiście lektura tego magazynu nie wystarczy, by zostać świetnym programistą Pythona – i nie taki jest jego cel. Mamy nadzieję, że po zapoznaniu się z treścią i zamieszczonymi przykładami zastosowań, zainspirujemy Czytelnika do dalszej nauki i eksperymentów z tym językiem.

Jednym z najbardziej rozpowszechnionych paradygmatów mijającej właśnie epoki IT jest arkusz kalkulacyjny, który już dawno przestał być zwykłą aplikacją, a stał się uniwersalnym narzędziem wykorzystywanym praktycznie wszędzie. Nie będzie przesadą stwierdzenie, że nie istnieje firma, która nie korzysta z arkusza kalkulacyjnego – trudno też znaleźć stanowisko biurowe (i nie tylko), na którym nie byłby wykorzystywany.

Jednak technologie informatyczne stale ewoluują. Dziś musimy sprawnie przetwarzać coraz większe ilości danych, które nie zawsze pasują do dwuwymiarowych tabel. Musimy tworzyć modele wykorzystujące coraz większą liczbę zmiennych. Choć model tabularyczny jest wciąż bardzo przydatny, jego przetwarzanie za pomocą arkusza kalkulacyjnego przestaje być wygodne – i to nie tylko ze względu na ogromną ilość danych. Z tych i innych powodów naukowcy i badacze na całym świecie, mimo że nie są programistami *sensu stricto*, zwracają się ku narzędziom takim jak Numpy czy Pandas, których ułamek możliwości tu prezentujemy.

Jeszcze dwadzieścia lat temu Python był uważany za język interpretowany, o wydajności z definicji niższej niż języki kompilowane, takie jak C czy C++. Dziś ten podział jest rozmyty: kiedy uruchamiamy kod Pythona, w praktyce jest on kompilowany, niektóre moduły są zaś tak zoptymalizowane, że nie ustępują wydajnością tym stosowanym w innych językach. Popularny niegdyś model rozwoju polegał na tym, że tworząc aplikację np. do rozpoznawania twarzy za pomocą OpenCV, opracowywano najpierw prototyp w Pythonie, ponieważ było to znacznie szybsze, prostsze i obarczone mniejszym ryzykiem wystąpienia błędów, a dopiero potem – po przetestowaniu prototypu – przechodzono do prac nad wersją produkcyjną w C++, by uzyskać odpowiednią wydajność. Dziś ta ostatnia faza jest często pomijana, ponieważ różnice w wydajności są w wielu przypadkach pomijalne, a tam, gdzie występuje wiele operacji wejścia/wyjścia, znikają całkowicie.

Na koniec ważna uwaga: ponieważ ze względu na edukacyjnych przykłady przedstawione są

w najprostszej i najkrótszej wersji, w wielu miejscach musieliśmy odejść od reguł sztuki. Po pierwsze, w kodzie stosowanym w praktyce wszystkie nazwy funkcji, zmiennych, klas itd. powinny być znaczące. Używanie jednoliterowych nazw zmiennych – poza specyficznymi przypadkami – to komplikowanie życia i sobie, i innym, którzy będą czytali nasz kod. Po drugie, pisząc programy w Pythonie, powinniśmy stosować się do specyfikacji przedstawionej w PEP8 [2] odnoszącej się do formatowania kodu. Ze względu na ograniczenia dotyczące miejsca w wielu przykładach byliśmy zmuszeni do złamania zawartych w nich reguł, nie ma jednak powodu, by się do nich nie stosować, pisząc własny kod. I po trzecie, jak krótko wspomnieliśmy przy okazji opisu błędów i wyjątków, pisząc prawdziwy kod, musimy zawsze sprawdzać wszelkie możliwe scenariusze niepowodzeń; jest to obowiązkowe zwłaszcza podczas operacji wejścia/wyjścia. Brak kontroli w tym przypadku to poważny błąd, który prędzej czy później doprowadzi do awarii programu. Jednak w przykładach musieliśmy zrezygnować z kontroli błędów, ponieważ kod musiałby kilkakrotnie zwiększyć swoją objętość.

Mamy nadzieję, że poznanie podstaw Pythona i jego przykładowych zastosowań przyniesie wszystkim Czytelnikom naszego magazynu ogromną satysfakcję. Jak wspominaliśmy, nie wszyscy muszą być programistami – jednak prędzej czy później znajomość tego języka programowania niewątpliwie przyda nam się w życiu.

INFO

[1] Indeks TIOBE: <https://www.tiobe.com/tiobe-index/>

[2] PEP 8: <https://peps.python.org/pep-0008/>

Linux Magazine jest miesięcznikiem specjalistycznym wydawanym na licencji Linux New Media USA, LLC, we współpracy z Computec Media GmbH, Fürth, Niemcy.

Wydawca Wiedza i Praktyka Sp. z o.o.

Redaktor Naczelny: Artur Skura, askura@linux-magazine.pl

Wydawca: Natalia Cybulska

Kierownik grupy tematycznej: Agata Jastrzębska

Korespondenci i współpracownicy:

Erik Bärawaldt, Chris Binnie, Zack Brown, Bruce Byfield, Karsten Günther, Marcel Hilzinger, Klaus Knopper, Christoph Langner, Jeff Layton, Martin Loschwitz, Patrick Neef, Dimitri Popov, Thorsten Scherf, Ferdinand Thommes

Opracowanie graficzne, skład i przygotowanie do druku

Raster studio, Norbert Bogajczyk, studio@rasterstudio.pl

Projekt okładki: Lori White

Reklama: reklama@linux-magazine.pl

Ceny prenumeraty łączonej (wersja papierowa i cyfrowa):

półroczna (6 numerów) 190 zł
roczna (12 numerów) 358,80 zł
dwuletnia (24 numery) 645,60 zł

Ceny e-prenumeraty:

półroczna (6 numerów) 130 zł
roczna (12 numerów) 289,80 zł
dwuletnia (24 numery) 526,60 zł

Ceny prenumeraty wersji drukowanej:

półroczna (6 numerów) 170 zł
roczna (12 numerów) 276 zł
dwuletnia (24 numery) 452 zł
Szczegóły: <http://linux-magazine.pl/zamow/subskrypcja>

Licencje korporacyjne, rozszerzone i niestandardowe

tel.: +48 22 429 43 05
e-mail: prenumerata@linux-magazine.pl

Zamówienia i obsługa prenumeraty:

tel.: +48 22 518 29 29
faks: +48 22 617 60 10
prenumerata@linux-magazine.pl

Linux Magazine

ul. Łotewska 9a, 03-918 Warszawa
www.linux-magazine.pl
tel.: +48 22 429 43 05, faks: +48 22 617 60 10

Wydawca dokłada wszelkich starań, aby publikowane w piśmie i na towarzyszących mu nośnikach informacje i oprogramowanie były poprawne i przydatne, jednakże Wydawca nie ponosi odpowiedzialności za efekty wykorzystania ich, w tym nie gwarantuje poprawnego działania programów.

Zawartość nośników CD-ROM i DVD jest sprawdzana oprogramowaniem antywirusowym przed rozpoczęciem procesu produkcji. Fizyczne uszkodzenia nośników należy zgłaszać do działu prenumeraty.

Żaden z materiałów opublikowanych w Linux Magazine nie może być powielany w jakiegokolwiek formie bez zgody Wydawcy. Właścicielem znaku towarowego Linux jest Linus Torvalds.

ISSN 1732-1263; Nakład 6000 egz.

Nr rejestrowy BDO: 000008579

PODSTAWY

- 6 Instalacja**
Aby móc zacząć programować w Pythonie, musimy najpierw pobrać i zainstalować interpreter tego języka, co zajmuje nie więcej niż kilka minut.
- 8 Środowisko programistyczne**
Domyślny wiersz poleceń Pythona i środowisko IDLE mogą się wydawać nieco spartańskie i nieprzystające do wymagań współczesnych użytkowników. Powstało więc wiele projektów z alternatywnymi środowiskami programistycznymi (IDE) umożliwiającymi łatwiejsze i bardziej wydajne tworzenie kodu w tym języku.
- 10 Zamiast kalkulatora**
Jednym z najprostszych sposobów korzystania z Pythona jest... użycie go w roli kalkulatora. Tak, ten potężny język programowania jest również szybką maszyną obliczeniową – w dodatku bardzo wygodną w obsłudze.
- 13 Wyświetlanie na ekranie**
Jedną z najbardziej podstawowych instrukcji Pythona jest `print()`, dzięki której wyświetlimy na terminalu dowolne dane.
- 16 Napisy**
Napisy, czyli łańcuchy znakowe, to jeden z podstawowych typów używanych w Pythonie, zatem umiejętność posługiwania się nimi jest absolutnie niezbędna dla każdego programisty Pythona.
- 19 Sekwencje**
Napisy to tzw. typy sekwencyjne, składające się z elementów ułożonych w określonym porządku.
- 21 Warunki**
Praktycznie każdy nietrywialny program w Pythonie zawiera instrukcje sterujące. Ich szczególnym przypadkiem są instrukcje warunkowe, takie jak `if` i `else`.
- 23 Listy**
Listy to uniwersalny typ danych składający się z elementów jednego lub różnych typów, przy czym znaczenie ma kolejność ich ułożenia.
- 27 Pętla `for`**
Pętla `for` jest podstawową konstrukcją iteracyjną w Pythonie.
- 29 Krotki**
Krotka to sekwencyjny typ danych, przypominający nieco listę, lecz w przeciwieństwie do niej niezmienny.
- 31 Słowniki**
Słowniki składają się z dwuelementowych par, z których jedna jest nazywana kluczem, a druga – wartością.

- 33 Zbiory**
Zbiór to typ danych składający się z elementów, które mogą być różnego typu, przy czym elementy te nie mogą się powtarzać.
- 35 Funkcje**
Funkcje umożliwiają wydzielenie logicznych fragmentów kodu do ponownego wykorzystania.
- 38 Fizzbuzz**
Pora na pierwsze wyzwanie programistyczne wykorzystujące przyswojoną do tej pory wiedzę.
- 40 Wyjątki**
Jeśli coś może pójść źle, to prędzej czy później to nastąpi. W Pythonie rozróżniamy dwa rodzaje takich sytuacji: błędy oraz wyjątki. Te pierwsze musimy naprawić, te drugie – obsłużyć.
- 42 Operacje na plikach**
Odczytywanie danych z plików i zapisywanie ich to podstawowa umiejętność programisty.
- 44 Moduły**
W Pythonie możemy korzystać z bogatych kolekcji funkcji znajdujących się w modułach, czyli bibliotekach funkcji.
- 46 Klasy**
Jednym z paradygmatów programowania obsługiwanych przez Pythona jest tzw. programowanie obiektowe. Bazuje ono na klasach posiadających własne atrybuty i metody.
- 48 Biblioteka standardowa**
Jedną z największych zalet Pythona jest rozbudowana i funkcjonalna biblioteka standardowa, umożliwiająca wykonanie wielu czynności bez konieczności importowania zewnętrznych modułów.

NAUKA O DANYCH

- 50 Bazy danych SQL**
W Pythonie możemy korzystać z uniwersalnego protokołu dostępu do wielu różnych baz danych; pokażemy, jak z niego korzystać na przykładzie komunikacji z SQLite.
- 52 Tablice, tablice...**
Każdy programista Pythona powinien znać podstawy pracy z tablicami NumPy. Jest to wysoce zoptymalizowany typ danych znajdujący się w module NumPy, umożliwiający tworzenie wielowymiarowych tablic z danymi numerycznymi i przeprowadzanie na nich wydajnych obliczeń.
- 54 Wykresy**
Python jest bardzo często wykorzystywany do przygotowywania wykresów. W praktyce do tego celu bardzo często jest wykorzystywana biblioteka Matplotlib.
- 56 Pandas i ramki danych**
Pandas to znakomita biblioteka Pythona służąca do przetwarzania i analizy danych. Swoje możliwości ujawnia zwłaszcza tam, gdzie są wykorzystywane dane tabularyczne.

OBRAZ I GRAFIKA

- 59 Grafika rastrowa**
Istnieje wiele metod obróbki obrazów rastrowych (bitmap) w Pythonie. Jedną z najpopularniejszych wiąże się z użyciem modułu Pillow, następcy popularnego niegdyś PIL-a.
- 62 G'MIC**
Prezentujemy przykład możliwości Pythona podczas obróbki zdjęć i pokazujemy, jak radzić sobie z nieobsługiwanymi platformami.
- 65 Rozpoznawanie obrazu**
Pokażemy, jak rozpocząć przygodę z fascynującym światem rozpoznawania obrazów dzięki bibliotece OpenCV.

INTERNET

- 71 Zapytania HTTP**
Komunikacja w sieci WWW odbywa się za pomocą żądań HTTP. W Pythonie znajdziemy wiele bibliotek do ich obsługi; jedną z najpopularniejszych jest `requests`.
- 73 Scraping**
Scraping to bezpośrednie pobieranie danych ze stron WWW w sposób zautomatyzowany, często bez pośrednictwa przeglądarki. W ten sposób uzyskujemy dostęp do różnego rodzaju informacji, które możemy na rozmaite sposoby przetwarzać.
- 75 Proste aplikacje webowe**
Python udostępni szereg potężnych narzędzi do tworzenia aplikacji webowych, w tym frameworki, takie jak Django i Flask. O ile Django jest bardzo rozbudowanym frameworkiem, Flask umożliwia szybkie i łatwe tworzenie nieskomplikowanych aplikacji, a także różnego rodzaju API z dostępem przez WWW.
- 78 Python a Slack**
Slack to popularne narzędzie do komunikacji i współpracy, używane przez wiele organizacji na całym świecie. Jeśli intensywnie korzystamy ze Slacka, zapewne docenimy fakt, że można go kontrolować za pomocą Pythona.
- 80 Automatykacja Facebooka**
Obsługa mediów społecznościowych to żmudne zadanie, zwłaszcza jeśli musimy umieszczać różne treści często i w wielu różnych sieciach. Na szczęście dzięki Pythonowi czynności te można do pewnego stopnia zautomatyzować.