LINUX MAGAZINE POLECA NR 1/2022

PYTHON

NAUKA PROGRAMOWANIA DLA KAŻDEGO

Poznaj Pythona od zera!

- Naucz się programować w najpopularniejszym języku świata!
- Poznaj podstawowe typy i struktury danych
- Naucz się sterować przepływem programu
- Poznaj najpopularniejsze moduły i biblioteki

Twórz własne aplikacje!

- Automatyzacja obsługi mediów społecznościowych
- Zaawansowana obróbka grafiki i efekty specjalne
- Podstawy nauki o danych
- Pandas zamiast Excela
- Własny bot Slacka
- Rozpoznawanie obrazów w OpenCV

Numer 1/2022
49,90 zł (8% vat)

ISSN 2720-5479

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 rozwijać 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 linijką. 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:micro, 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 multimediów 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 względów 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 kilkukrotnie 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 Bärwaldt, 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) **358,80zł**
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

ISSN 1732-1263; Nakład 6000 egz.

Nr rejestrowy BDO: 000008579

PODSTAWY

NAUKA O DANYCH

OBRAZ I GRAFIKA

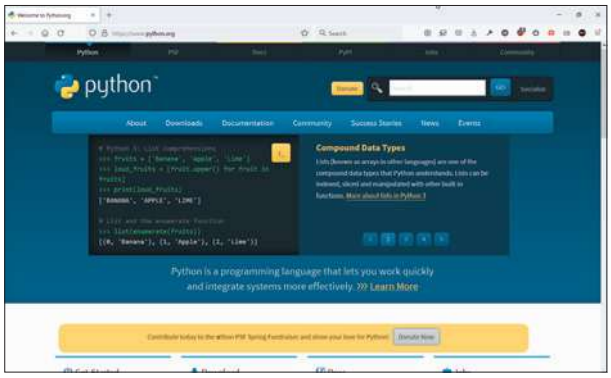
INTERNET

Instalacja

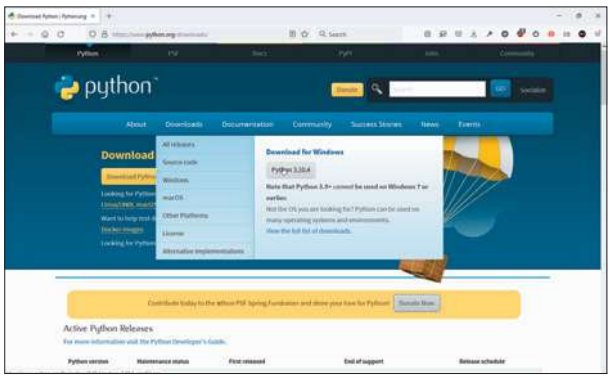
Aby móc zacząć programować w Pythonie, musimy najpierw pobrać i zainstalować interpreter tego języka, co zajmuje nie więcej niż kilka minut.

Windows

1. Wchodzimy na witrynę python.org

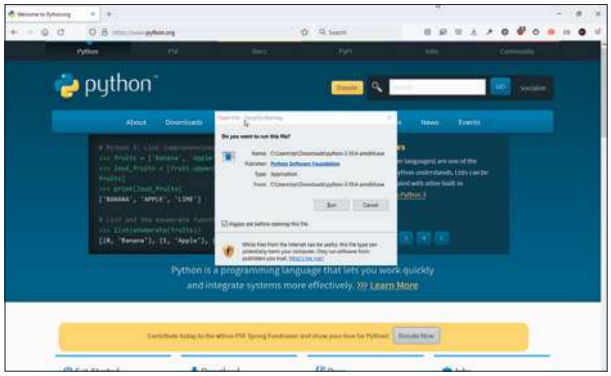


2. Klikamy zakładkę *Download* i duży przycisk z napisem Python oraz numerem wersji – jest to najnowsza stabilna wersja Pythona.



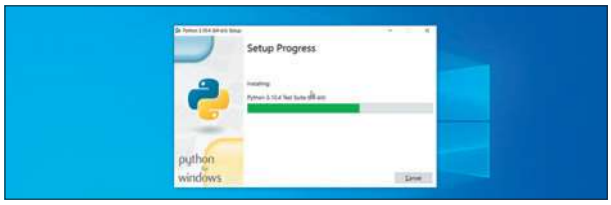
3. Klikamy pobrany plik wykonywalny.

4. Windows może nas zapytać, czy na pewno chcemy go uruchomić – potwierdzamy.



5. Pojawi się okno instalatora – zatwierdzamy domyślne ustawienia i klikamy *Install now*.

6. Instalator rozpocznie pracę, kopiując niezbędne pliki i przeprowadzając wstępną konfigurację.



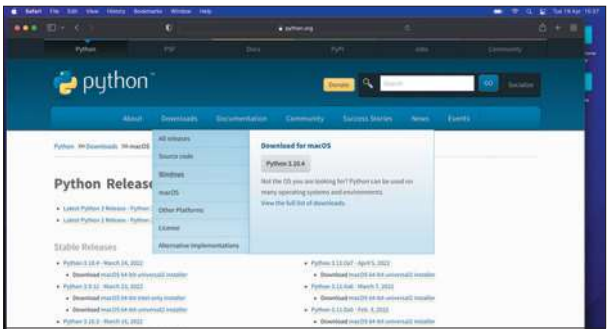
7. Po zakończeniu całego procesu powinniśmy ujrzeć ekran informujący nas, że cały proces przebiegł pomyślnie.

8. Python został zainstalowany – w menu startowym znajdziemy dwie pozycje: Python oraz IDLE. Ta pierwsza to wiersz poleceń Pythona, czyli okno, w którym możemy wpisywać polecenia Pythona, interpreter je od razu wykona. Natomiast IDLE to nieco bardziej rozbudowane środowisko, w którym możemy nie tylko wpisywać wiersz po wierszu kod Pythona, ale również tworzyć skrypty napisane w tym języku, edytować je i uruchamiać.

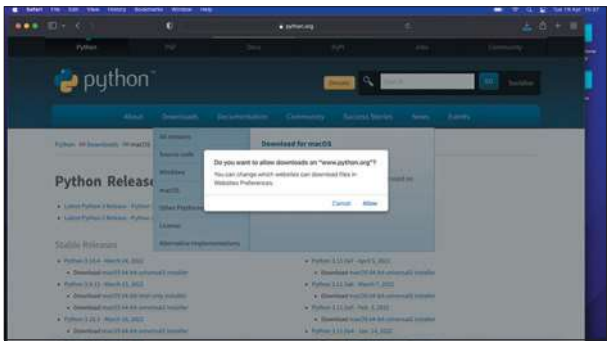
macOS

Do niedawna do macOS-a dołączana była starsza wersja Pythona (z linii 2.x), która nie jest zbyt przydatna do codziennej pracy, zaś w Monterey Apple ostatecznie usunął ją z systemu. Dlatego najprościej jest zainstalować najnowszą wersję Pythona bezpośrednio z witryny projektu.

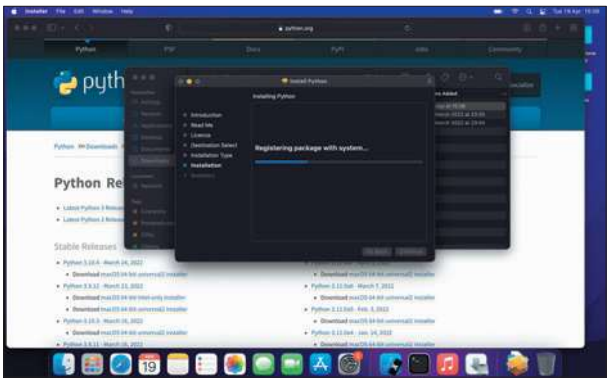
W tym celu wchodzimy na stronę Python.org i klikamy zakładkę *Downloads* – witryna powinna automatycznie rozpoznać nasz system i zaproponować pobranie najnowszego wydania Pythona.



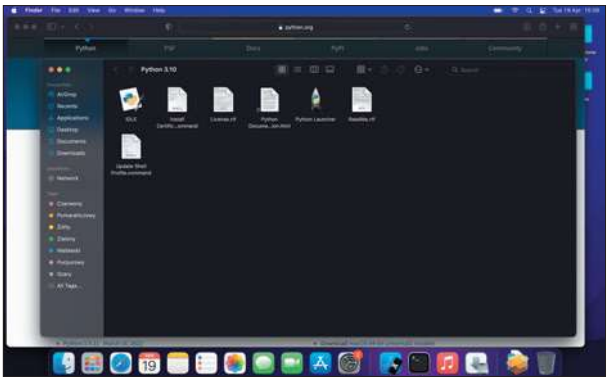
Może być konieczne zezwolenie na pobieranie plików z witryny Python.org.



Następnie instalator skopiuje wszystkie wymagane pliki w odpowiednie miejsca i przeprowadzi pozostałe wymagane czynności.



Po instalacji ujrzymy folder, w którym znajdziemy IDLE oraz Python Launcher – niewielkie narzędzie, które ułatwia konfigurację uruchamiania skryptów Pythona podwójnym kliknięciem.



Uwaga: jeśli używamy jednej z wcześniejszych wersji macOS-a lub aktualizowaliśmy go, w systemie nadal może być obecny starszy Python 2.7 instalowany przez Apple. Możemy to łatwo sprawdzić, wydając w Terminalu polecenie *python* i sprawdzając numer wersji. Jeśli jest to 2.7, powinniśmy uruchamiać Pythona poleceniem *python3*.

Linux

Dobra wiadomość: obecnie Python instalowany jest domyślnie w wielu popularnych dystrybucjach Linuksa – najprawdopodobniej więc nie musimy w ogóle nic robić. Jeśli jednak z jakiegoś powodu nie znajdziemy go w systemie, wystarczy zainstalować go za pomocą menedżera pakietów, tak jak dowolny inny program. Np. w Ubuntu zrobimy to poleceniem *sudo apt install python*.