LINUX MAGAZINE POLECA NR 2/2023

PYTHON

NAUKA PROGRAMOWANIA DLA KAŻDEGO

OPANUJ PYTHONA OD PODSTAW:
od procesu instalacji po korzystanie z biblioteki standardowej

Tworzenie gier w Pythonie: **PONG, TETRIS I SPACE INVADERS!**

Generowanie grafiki AI z poziomu **PYTHONA**

PYXEL: Kombajn do tworzenia gier retro

Aplikacja mobilna z backendem w **PYTHONIE**

FULL STACK: Kompletna aplikacja webowa krok po kroku

ALTERNATYWNE ŚRODOWISKA PROGRAMISTYCZNE

KLUCZOWE UMIEJĘTNOŚCI: napisy i sekwencje

Własny **CHATGPT** w Pythonie

DLA POCZĄTKUJĄCYCH:

Sterowanie przebiegiem programu: **INSTRUKCJE WARUNKOWE I PĘTLE**

Efektywne zarządzanie kodem: **FUNKCJE I WYJĄTKI**

Operacje na plikach: **ODCZYT I ZAPIS DANYCH**

Pierwsze wyzwanie programistyczne: **FIZZBUZZ**

STRUKTURY DANYCH: krotki, słowniki i zbiory

Numer 2/2023
49,90 zł (8% vat)
ISSN 2720-5479

Pracujesz w środowisku linuksowym?

- Chcesz być na bieżąco z nowościami? Potrzebujesz wsparcia?
- A może szukasz łatwego i stałego dostępu do aktualnych treści związanych z Linuksem i technologiami open source?

Zamów już dziś prenumeratę światowego bestsellera „Linux Magazine" i miej pewność, że najnowszy numer trafi prosto do Ciebie.

Z nami poszerzysz swoje zawodowe kompetencje:

Skonfigurujesz.
Skonfigurujesz serwery udostępniające różnego rodzaju usługi

Poznasz.
Poznasz nowoczesne rozwiązania służące do monitorowania systemów i sieci.

Lepiej zrozumiesz.
Lepiej zrozumiesz działanie Systemd i innych kluczowych komponentów systemu.

Zapoznasz się.
Zapoznasz się z najciekawszymi i najnowszymi rozwiązaniami open source.

Będziesz na bieżąco.
Będziesz na bieżąco z zagrożeniami i metodami zabezpieczania się przed nimi.

Dowiesz się.
Dowiesz się jak zoptymalizować funkcjonowanie systemu pod kątem wydajności.

Nauczysz się.
Nauczysz się jak działają łańcuchy Markowa i w jaki sposób uczy się sztuczna inteligencja.

LINUX MAGAZINE

Dołącz do grona prenumeratorów Linux Magazine i korzystaj z atrakcyjnych zniżek i promocji dostępnych dla Czytelników.

Skontaktuj się z naszym specjalistą, który zaproponuje Ci najlepszą ofertę prenumeraty: **+48 22 518 29 29** lub zajrzyj na naszą stronę: **https://linux-magazine.pl/**

Co w numerze?

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

Kiedyś, kiedy każdy mikrokomputer był wyposażony w interpreter języka programowania, listingi zaś mozolnie przepisywało się z magazynów komputerowych znak po znaku, nierzadko klnąc pod nosem z powodu drobnych błędów, programowanie było czymś, z czym chcąc nie chcąc, stykał się każdy użytkownik komputera – choćby po to, by móc załadować grę. Nierzadko jednak zainteresowanie grami przekształcało się w chęć dowiedzenia się, jak to wszystko działa – np. po to, by móc tworzyć własne gry. Nie było to jednak proste, trzeba było bowiem całą pracę wykonać samemu, dostępne zaś źródła były dość skąpe.

Dziś sytuacja wygląda diametralnie inaczej. Do dyspozycji mamy potężne narzędzia, takie jak Unity czy Unreal Engine, które umożliwiają tworzenie zaawansowanych gier przewyższających grafiką to, co można było sobie wyobrazić w latach 80. czy nawet 90. ubiegłego wieku. A mimo to obserwujemy wzrost popularności gier 2D takich jak Deltarune czy Celeste, które wręcz epatują swoją surową „pikselowością", udowadniając, że hiperrealizm to niejedyny kierunek rozwoju współczesnych gier.

Niniejsze wydanie „Linux Magazine Poleca", podobnie jak poprzednie, składa się z dwóch części. Pierwsza to wprowadzenie do języka Python – niezbędne, by móc zrozumieć kolejne artykuły. Omawiamy poszczególne zagadnienia krok po kroku, ilustrując każde przykładami w taki sposób, by wyeliminować wszystkie możliwe wątpliwości. Jednocześnie zachęcamy Czytelników do przetestowania zdobytej wiedzy na różnych platformach online, takich jak Codewars, by stale pogłębiać swoją znajomość języka jako takiego oraz algorytmiki. Dotyczy to zwłaszcza tych wszystkich, którzy myślą o karierze programisty.

Druga część magazynu poświęcona jest zagadnieniom praktycznym i składa się z 3 sekcji. Zaczynamy od tworzenia gier, wykorzystując proste w użyciu biblioteki, takie jak Pygame czy Pyxel. Pokażemy, jak o własnych siłach stworzyć kultowe gry, które towarzyszyły pokoleniu młodych graczy kilkadziesiąt lat temu – Pong, Tetris i Space Invaders.

W drugiej sekcji zajmiemy się programowaniem backendu. Ponieważ Python nie istnieje w próżni, pokażemy, jak w praktyce odbywa się interakcja między pythonowym backendem (Flask) a javascriptowym frontendem (React). Następnie pójdziemy o krok dalej i rozwiniemy ten sam paradygmat na aplikacje mobilne, wykorzystując Flaska do zbudowania backendu aplikacji mobilnej.

Wreszcie w ostatniej sekcji skupiamy się na coraz bardziej popularnej ostatnio sztucznej inteligencji, a zwłaszcza na jej dwóch aspektach: generowaniu tekstu i obrazków. Pokażemy między innymi, jak z poziomu Pythona komunikować się z API OpenAI, tworząc własnego klona ChatGPT oraz w jaki sposób wygenerować obrazy powojennej Warszawy pędzlem wielkich malarzy.

Mamy nadzieję, że w tym numerze każdy Czytelnik znajdzie coś dla siebie, a zamieszczone tu przykłady posłużą za inspirację do stworzenia własnych, bardziej zaawansowanych i rozbudowanych projektów. ■■■

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.

SPIS TREŚCI

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: Patrycja Oleksiejuk

Kierownik grupy tematycznej: Szymon Danowski

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: Magdalena Huta

Reklama: *reklama@linux-magazine.pl*

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

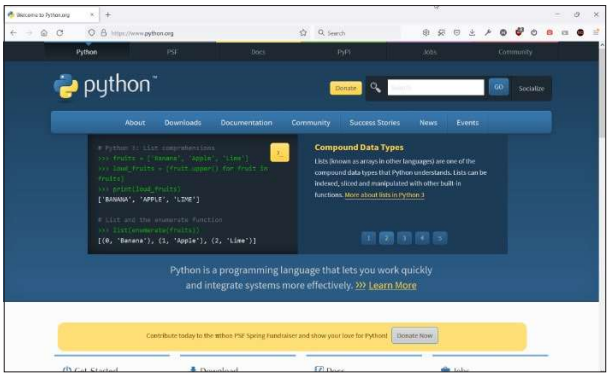
PODSTAWY

Instalacja

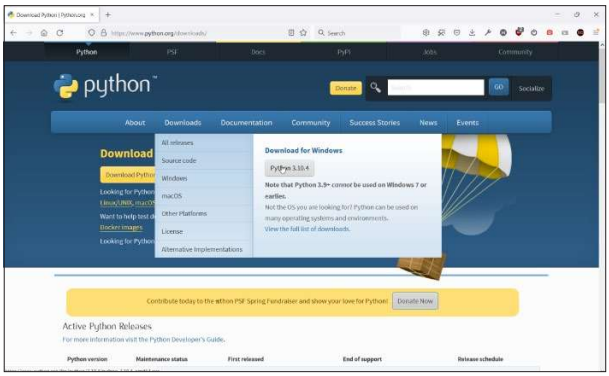
Aby móc zacząć programować w Pythonie, musimy najpierw pobrać i zainstalować interpreter tego języka, co zajmuje nie więcej niż kilka minut.

Windows

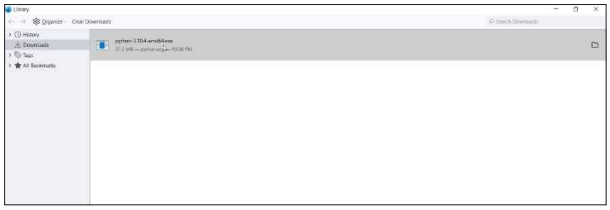
1. Wchodzimy na witrynę python.org



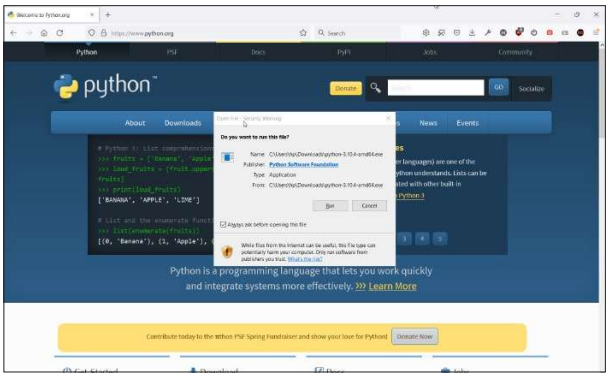
2. Klikamy zakładkę *Download* i duży przycisk z napisem Python oraz numerem wersji – jest to najnowsza stabilna wersja Pythona.



3. Klikamy pobrany plik wykonywalny.

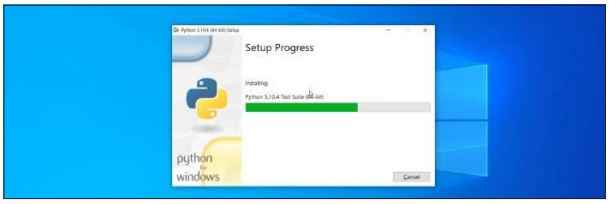


4. Windows może nas zapytać, czy na pewno chcemy go uruchomić – potwierdzamy.



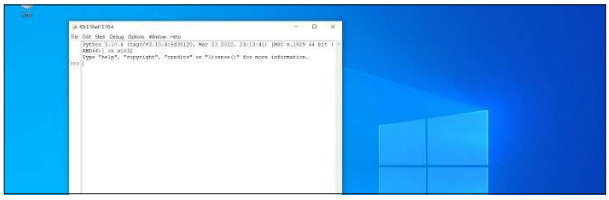
5. Pojawi się okno instalatora – zatwierdzamy domyślne ustawienia i klikamy *Install now*.

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.

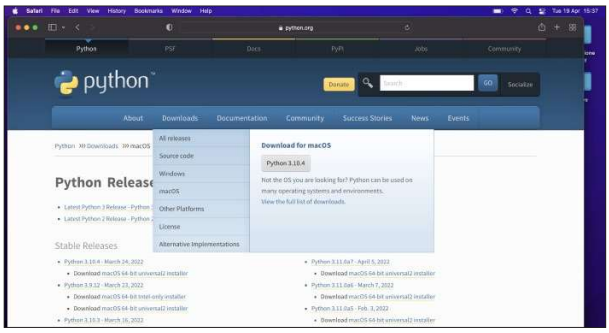
8. Python został zainstalowany – w menu startowym znajdziemy dwie pozycje: Python oraz IDLE. Ta pierwsza to wiersz poleceń Pythona, czyli okno, w którym możemy wpisywać polecenia Pythona, interpreter je od razu wykona. Natomiast IDLE to nieco bardziej rozbudowane środowisko, w którym możemy nie tylko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



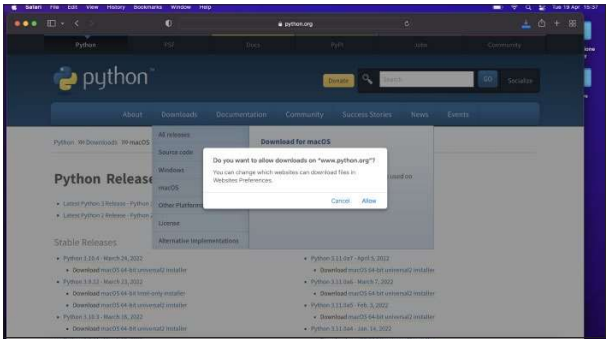
macOS

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.

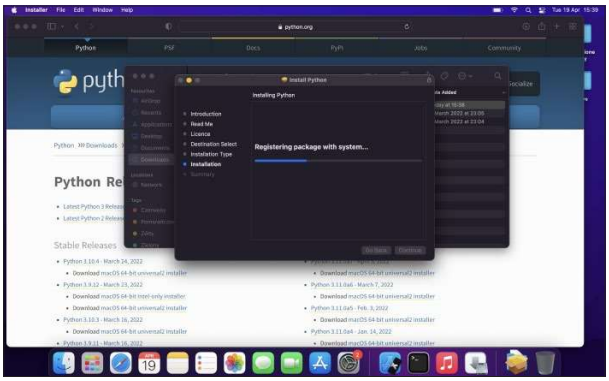
W tym celu wchodzimy na stronę Python.org i klikamy zakładkę *Downloads* – witryna powinna automatycznie rozpoznać nasz system i zaproponować pobranie najnowszego wydania Pythona.



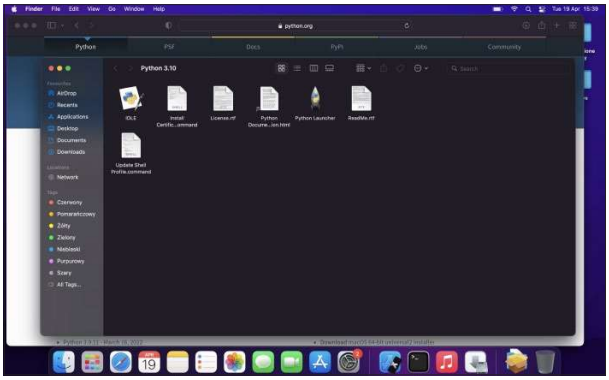
Może być konieczne zezwolenie na pobieranie plików z witryny Python.org.



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*.