

---

# PYTHON DLA NAJMŁODSZYCH W 1 DZIEŃ

---

NAUCZ SIĘ KODOWAĆ W PYTHONIE JUŻ W 12 GODZIN

## SPIS TREŚCI

---

|                                                                 |                                         |
|-----------------------------------------------------------------|-----------------------------------------|
| Rozdział 1: Powitajmy się z Pythonem! .....                     | 5                                       |
| Co to jest Python .....                                         | 6                                       |
| Historia Pythona i jego zastosowanie .....                      | 9                                       |
| Pythonowe ciekawostki.....                                      | 11                                      |
| Podstawowe komendy i interakcja z Pythonem.....                 | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 2: Pierwsze kroki – instalacja i środowisko .....      | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Przygotowanie komputera do pracy – pobieranie Pythona .....     | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Instalacja Pythona krok po kroku...                             | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Tworzenie pierwszego środowiska programistycznego .....         | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Wprowadzenie do IDLE i powłoki Pythona.....                     | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 3: Twoje pierwsze programy – „Cześć, Świecie!” .....   | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Jak napisać pierwszy program w Pythonie.....                    | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Zrozumienie składni Pythona na przykładzie prostego skryptu.... | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Uruchamianie i debugowanie pierwszego programu .....            | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Tworzenie interaktywnego skryptu z użytkownikiem.....           | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 4: Zabawa z żółwiem – poznajemy Turtle Graphics.....   | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Wprowadzenie do modułu Turtle ..                                | <b>Błąd! Nie zdefiniowano zakładki.</b> |

|                                                              |                                         |
|--------------------------------------------------------------|-----------------------------------------|
| Rysowanie prostych kształtów i linii .....                   | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Używanie pętli do tworzenia wzorów .....                     | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Personalizacja żółwia – kolory i długopis .....              | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 5: Podstawy Pythona – zmienne i typy danych.....    | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Definiowanie zmiennych i ich nazewnictwo....                 | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Typy danych w Pythonie: liczby, łańcuchy znaków, listy ..... | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Operacje na zmiennych i konwersja typów.....                 | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Proste operacje matematyczne i ich zastosowanie .....        | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 6: Przerwa na lunch i krzyżówka z Pythonem .....    | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Propozycje zdrowego lunchu dla młodego programisty .....     | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozwiązywanie krzyżówki z terminami związanymi z Pythonem    | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Krótkie zadania logiczne do rozwiązania podczas przerwy..... | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Odpoczynek i przygotowanie do kolejnych wyzwań.....          | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Rozdział 7: Sterowanie przepływem – instrukcje warunkowe.... | <b>Błąd! Nie zdefiniowano zakładki.</b> |
| Co to jest instrukcja warunkowa i jak ją używać.....         | <b>Błąd! Nie zdefiniowano zakładki.</b> |

Zadania z użyciem instrukcji warunkowych.....**Błąd! Nie zdefiniowano zakładki.**

Tworzenie prostych gier logicznych.....**Błąd! Nie zdefiniowano zakładki.**

Rozdział 8: Pętle i powtórzenia – for i while w akcji .....**Błąd! Nie zdefiniowano zakładki.**

Zrozumienie pętli for i while ..... **Błąd! Nie zdefiniowano zakładki.**

Praktyczne ćwiczenia z wykorzystaniem pętli**Błąd! Nie zdefiniowano zakładki.**

Kiedy używać której pętli – for czy while?.....**Błąd! Nie zdefiniowano zakładki.**

Infinite loop - co to jest i jak go unikać .....**Błąd! Nie zdefiniowano zakładki.**

Rozdział 9: Zabawy z listami – twoja pierwsza kolekcja .....**Błąd! Nie zdefiniowano zakładki.**

Tworzenie i modyfikacja list..... **Błąd! Nie zdefiniowano zakładki.**

Dostęp do elementów listy i ich indeksowanie .....**Błąd! Nie zdefiniowano zakładki.**

Metody list: sortowanie, dodawanie, usuwanie elementów ....**Błąd! Nie zdefiniowano zakładki.**

Zastosowanie list w prostych algorytmach .....**Błąd! Nie zdefiniowano zakładki.**

Rozdział 10: Funkcje w Pythonie – piszemy własne skrypty .....**Błąd! Nie zdefiniowano zakładki.**

Definiowanie funkcji i ich struktura .....**Błąd! Nie zdefiniowano zakładki.**

Argumenty funkcji i wartości zwracane .....**Błąd! Nie zdefiniowano zakładki.**

Rekurencja i kiedy z niej korzystać **Błąd! Nie zdefiniowano zakładki.**

Organizacja kodu za pomocą funkcji.....**Błąd! Nie zdefiniowano zakładki.**

Rozdział 11: Mały projekt – stwórz swoją grę w Pythonie .....**Błąd! Nie zdefiniowano zakładki.**

Planowanie prostego projektu gry. **Błąd! Nie zdefiniowano zakładki.**

Implementacja logiki gry krok po kroku .....**Błąd! Nie zdefiniowano zakładki.**

Testowanie i debugowanie projektu.....**Błąd! Nie zdefiniowano zakładki.**

Dodawanie elementów interaktywnych i graficznych.....**Błąd! Nie zdefiniowano zakładki.**

Rozdział 12: Co dalej z Pythonem? ..... **Błąd! Nie zdefiniowano zakładki.**

Jak kontynuować naukę Pythona po tym kursie .....**Błąd! Nie zdefiniowano zakładki.**

Dodatkowe zasoby i miejsca w sieci do nauki programowania..... **Błąd! Nie zdefiniowano zakładki.**

Inspirujące projekty, które można zrealizować w Pythonie ....**Błąd! Nie zdefiniowano zakładki.**

Podsumowanie i cele na przyszłość .....**Błąd! Nie zdefiniowano zakładki.**

# ROZDZIAŁ 1: POWITAJMY SIĘ Z PYTHONEM!

---

## CO TO JEST PYTHON

---

Python to dynamiczny, interpretowany język programowania wysokiego poziomu, który stał się jednym z najbardziej popularnych języków programowania na świecie. Został stworzony przez Guido van Rossuma i po raz pierwszy wydany w 1991 roku. Jego filozofia projektowa akcentuje czytelność kodu i syntaksję, która umożliwia programistom pisanie programów w mniejszej liczbie linii kodu niż to możliwe w językach takich jak C++ czy Java. Python wspiera różne paradygmaty programowania, w tym obiektowe, imperatywne oraz w pewnym zakresie funkcjonalne.

Jedną z głównych zalet Pythona jest jego prosta składnia, która przypomina pseudokod. To sprawia, że jest on znakomitym językiem dla początkujących programistów, ponieważ mogą oni skupić się na nauce podstawowych koncepcji programowania bez zmagania się z zawiłościami składniowymi. W przeciwieństwie do wielu języków, Python wymaga mniejszej ilości kodu do wykonania tych samych zadań, co sprzyja szybkiemu i efektywnemu tworzeniu oprogramowania.

Interaktywny tryb Pythona, nazywany także REPL (Read-Eval-Print Loop), pozwala na szybkie eksperymentowanie i testowanie fragmentów kodu, co jest idealne dla początkujących, którzy mogą natychmiast zobaczyć wyniki swojej pracy. Dodatkowo, język ten posiada bardzo rozbudowaną standardową bibliotekę, która zawiera moduły i funkcje wspierające wiele codziennych zadań programistycznych, od operacji na plikach, przez komunikację sieciową, aż po analizę danych.

Python jest również niezwykle wszechstronny, co widać po jego zastosowaniach w różnych dziedzinach, takich jak rozwój stron internetowych, analiza danych, uczenie maszynowe, sztuczna inteligencja, nauki przyrodnicze, a nawet tworzenie gier. Jego skuteczność w analizie danych i uczeniu maszynowym jest wspierana przez potężne biblioteki, takie jak NumPy, Pandas, Matplotlib, Scikit-learn i TensorFlow, które uczyniły Python standardem w obszarach big data i data science.

Dodatkowo, Python jest językiem wieloplatformowym, co oznacza, że programy napisane w Pythonie mogą być uruchamiane na różnych systemach operacyjnych, takich jak Windows, macOS i Linux, bez konieczności modyfikacji kodu. Ta przenośność w połączeniu z łatwością wdrażania aplikacji sprawia, że Python jest atrakcyjnym wyborem dla firm i programistów pragnących dotrzeć do szerokiego spektrum użytkowników.

Popularność Pythona doprowadziła do powstania ogromnej społeczności, która nieustannie wspiera i rozwija język, tworząc liczne moduły, wtyczki i narzędzia, które są dostępne dla wszystkich. Jest to język o otwartym kodzie źródłowym, co oznacza, że każdy może przyczynić się do jego rozwoju oraz korzystać z pracy innych. To otwarte środowisko sprzyja współdzieleniu wiedzy i współpracy, co czyni naukę Pythona jeszcze bardziej atrakcyjną, szczególnie dla osób, które chcą rozwijać swoje umiejętności w społecznościowym kontekście.

W środowisku edukacyjnym Python jest często wybierany jako pierwszy język programowania ze względu na swój łatwy do zrozumienia składnik, a także przez wzgląd na szerokie możliwości, jakie oferuje początkującym programistom. Ucząc się Pythona, młodzi adepci programowania mogą z łatwością przechodzić od prostych skryptów do bardziej złożonych aplikacji, zdobywając niezbędne umiejętności, które posłużą im w dalszej karierze zawodowej.

W kontekście kariery zawodowej, znajomość Pythona jest wysoko ceniona na rynku pracy. Programiści Pythona są poszukiwani w wielu sektorach przemysłu ze względu na wszechstronność i potencjał języka w nowoczesnych technologiach. Python jest często wykorzystywany w startupach technologicznych i przez duże korporacje, co czyni umiejętności w Pythonie atrakcyjnymi dla pracodawców i otwiera drzwi do wielu możliwości zawodowych.

Pod względem technicznym, Python ma dynamiczne typowanie, co oznacza, że nie musimy z góry deklarować typów danych. Pozwala to na większą elastyczność w kodowaniu i skraca czas potrzebny na prototypowanie rozwiązań. Mimo że dynamiczne typowanie może prowadzić do pewnych błędów w czasie wykonywania programu,



środowisko Pythona dostarcza narzędzi, które pomagają w ich wykrywaniu i rozwiązywaniu.

Wprowadzenie do Pythona często zaczyna się od nauczania podstawowych struktur takich jak listy, słowniki, pętle i funkcje, które są podstawą większości programów. Zrozumienie i opanowanie tych elementów stanowi solidne podstawy do dalszej nauki i eksploracji bardziej zaawansowanych aspektów programowania. Co ważne, dzięki przejrzystemu i uporządkowanemu stylowi kodowania, Python jest również bardzo dobrym językiem do nauczania dobrych praktyk programistycznych, które są ważne w każdym języku programowania.

Python jest również znany ze swojej możliwości integracji z innymi językami i technologiami. Można łatwo połączyć Pythona z kodem napisanym w C, C++ czy Java, co umożliwia wykorzystanie gotowych rozwiązań oraz optymalizację wydajności tam, gdzie jest to wymagane. Ta zdolność do integracji sprawia, że Python jest często używany jako "klej" łączący różne komponenty systemu.

Wnioskując, nauka Pythona oferuje wiele korzyści zarówno dla początkujących programistów, jak i dla doświadczonych deweloperów, którzy poszukują uniwersalnego języka, który łączy prostotę z potężnymi możliwościami. Jego łatwa do nauki składnia, wszechstronność i przenośność, a także wsparcie społeczności i wszechobecność w przemyśle technologicznym sprawiają, że Python jest doskonałym wyborem dla tych, którzy chcą rozpocząć lub rozwijać swoją karierę w programowaniu.

## HISTORIA PYTHONA I JEGO ZASTOSOWANIE

---

Python, język programowania stworzony przez Guido van Rossuma w późnych latach 80-tych, wyrosnął na jednego z gigantów w świecie technologii. Jego historia rozpoczęła się w Centrum Matematyki i Informatyki w Holandii, gdzie van Rossum pracował nad systemem operacyjnym Amoeba. Z potrzeby posiadania łatwego w użyciu i rozszerzalnego języka skryptowego, van Rossum postanowił stworzyć Pythona w grudniu 1989 roku. Pierwsza publiczna wersja, Python 0.9.0, została wydana w lutym 1991 roku. Cechowała ją już wtedy jasność składni i czytelność, które stały się znakiem rozpoznawczym Pythona.

Z czasem Python ewoluował, stając się wszechstronnym narzędziem programistycznym. Przejście na Python 2.0 w 2000 roku przyniosło wiele ulepszeń, w tym wsparcie dla Unicode i pełniejszą kolekcję modułów standardowych. W 2008 roku, z kolei, narodził się Python 3.0, znany również jako "Python 3000" lub "Py3k", który miał być ulepszoną wersją języka, nawet za cenę bycia niekompatybilnym wstecz z poprzednimi wersjami. To przejście, chociaż początkowo spotkało się z oporem ze strony społeczności, okazało się ważnym krokiem w dążeniu do większej klarowności i spójności języka.

Rozwój Pythona nie byłby możliwy bez aktywnej i zaangażowanej społeczności. Język rozwijany jest przez Python Software Foundation, organizację non-profit, która zajmuje się zarówno rozwojem języka, jak i organizacją konferencji oraz wsparciem dla społeczności programistów. Społeczność Pythona słynie ze swojej otwartości i pomocności, a także z organizowania licznych wydarzeń na całym świecie, takich jak warsztaty, meetupy czy konferencje, gdzie początkujący programiści mogą uczyć się od doświadczonych mentorów.

W dzisiejszym świecie Python znajduje zastosowanie w wielu różnych dziedzinach. Jednym z głównych obszarów, w których Python jest wykorzystywany, są aplikacje webowe. Frameworki takie jak Django czy Flask sprawiają, że tworzenie skomplikowanych stron internetowych i aplikacji webowych jest szybsze i bardziej intuicyjne. Język ten jest również dominującym narzędziem w dziedzinie nauki o danych, uczenia maszynowego i sztucznej inteligencji. Biblioteki takie jak NumPy,

pandas, TensorFlow i scikit-learn są nieocenione dla analityków danych i inżynierów AI, umożliwiając im przetwarzanie ogromnych zbiorów danych i budowanie zaawansowanych modeli.

Dodatkowo, Python odgrywa znaczącą rolę w automatyzacji i zarządzaniu systemami. Jego skryptowy charakter sprawia, że jest idealny do pisania skryptów, które automatyzują rutynowe zadania administracyjne, testowania oprogramowania oraz zarządzania infrastrukturą sieciową. W branży gier Python również ma swoje miejsce, zarówno w prototypowaniu, jak i w produkcji komercyjnej. Pygame to biblioteka, która dostarcza funkcjonalności potrzebne do tworzenia gier, a wiele profesjonalnych narzędzi do tworzenia gier, jak na przykład Blender, wykorzystuje Pythona jako jeden z języków skryptowych.

Edukacja to kolejny sektor, w którym Python zdobył popularność. Jego prostota i przejrzystość czynią go idealnym językiem do nauki programowania, zarówno dla dzieci, jak i dorosłych. Wiele szkół i uniwersytetów na całym świecie wprowadziło Pythona jako język wprowadzający do informatyki, co świadczy o jego znaczeniu jako narzędzia edukacyjnego.

W dziedzinie zastosowań naukowych, Python jest szeroko wykorzystywany przez naukowców i inżynierów do symulacji, obliczeń statystycznych oraz analizy danych. Język ten znajduje zastosowanie w takich dziedzinach jak astronomia, biologia, chemia, fizyka i geoinformatyka. Dzięki bibliotekom takim jak SciPy, matplotlib i Biopython, badacze mogą wykonywać zaawansowane analizy i wizualizacje danych, co przyspiesza proces odkrywania nowych wiedzy.

W biznesie, Python jest wykorzystywany do budowy systemów CRM, systemów zarządzania zasobami przedsiębiorstw (ERP), a także w finansach do analizy rynków, algorytmicznego handlu czy wykrywania oszustw. Język ten jest ceniony za szybkość tworzenia prototypów i możliwość szybkiego wprowadzania zmian, co jest kluczowe w dynamicznie zmieniających się środowiskach biznesowych.

W sektorze bezpieczeństwa komputerowego Python również znalazł swoje miejsce. Jest wykorzystywany do tworzenia narzędzi do testowania penetracyjnego, analizy malware i monitoringu sieci, co jest istotne dla ochrony przed cyberatakami. Narzędzia takie jak Scapy czy sqlmap są pisane w Pythonie i stanowią standardowe wyposażenie specjalistów ds. bezpieczeństwa IT.

Na koniec warto zauważyć, że Python stał się kluczowym elementem w rozwijających się technologiach, takich jak Internet Rzeczy (IoT) oraz wytwarzaniu oprogramowania dla urządzeń embedded. Python zapewnia platformę, na której można szybko tworzyć i wdrażać aplikacje, co jest kluczowe przy pracy z ogromną ilością urządzeń i danych generowanych przez IoT.

Historia Pythona jest dowodem na to, jak elastyczny i potężny może być język programowania, gdy wspierany jest przez społeczność i dostosowywany do zmieniających się potrzeb. Jego rola w rozwoju technologicznym jest nie do przecenienia, a różnorodność jego zastosowań świadczy o tym, że Python pozostanie istotnym narzędziem dla programistów, naukowców, inżynierów i biznesmenów jeszcze przez wiele lat.

---

## PYTHONOWE CIEKAWOSTKI

---

Python to nie tylko potężne narzędzie dla programistów na całym świecie, ale także źródło wielu fascynujących ciekawostek, które mogą zainteresować najmłodszych adeptów programowania. Jedną z takich ciekawostek jest fakt, że Python został nazwany na cześć brytyjskiej grupy komediowej Monty Python. Twórca języka, Guido van Rossum, był wielkim fanem serialu "Latający cyrk Monty Pythona" i chciał, aby praca z jego językiem była zabawna i nieprzemijająca – podobnie jak skecze swoich ulubionych komików.

Zastanawiając się nad tym, jak Python może zachęcić dzieci do nauki programowania, warto zwrócić uwagę na prostotę jego składni. Python został zaprojektowany z myślą o czytelności kodu, co oznacza, że nawet

początkujący mogą szybko zrozumieć, co dana linijka kodu robi. Ta cecha Pythona sprawia, że jest on idealnym językiem dla dzieci, które dopiero rozpoczynają swoją przygodę z programowaniem. Na przykład, w Pythonie pętla `for` wykorzystywana do iteracji przez różne obiekty jest intuicyjna i nie wymaga wielu skomplikowanych konstrukcji, jak w innych językach.

Jednym z aspektów Pythona, który może wywołać uśmiech na twarzy każdego młodego programisty, jest obecność tzw. "easter eggs" (wielkanocnych jajek) – niespodzianek ukrytych przez programistów w języku. Na przykład, jeśli w interaktywnej konsoli Pythona wpisze się `'import this'`, wyświetlany jest tajemniczy 'The Zen of Python' – zbiór aforystycznych zasad dobrego programowania autorstwa Tima Petersa, które w humorystyczny sposób podsumowują filozofię projektowania języka.

W świecie Pythona znajdziemy również bogactwo bibliotek, które umożliwiają tworzenie gier, animacji czy nawet prostych robotów. Jedną z takich bibliotek jest Pygame, która pozwala na tworzenie gier z wykorzystaniem Pythona. Dzięki niej, nawet młodzi programiści mogą stworzyć własną grę komputerową, ucząc się przy tym logicznego myślenia, rozwiązywania problemów i podstaw matematyki. Warto podkreślić, że Pygame jest dostępna do pobrania za darmo, co obniża barierę wejścia dla młodych entuzjastów kodowania.

Ciekawą cechą Pythona, którą mogą docenić najmłodsi, jest fakt, że Python jest używany do nauczania komputerów, jak rozpoznawać ludzki język, co jest dziedziną zwaną przetwarzaniem języka naturalnego (NLP). Python, dzięki bibliotece Natural Language Toolkit (NLTK), umożliwia tworzenie programów, które potrafią analizować, rozumieć i nawet generować ludzki język. To otwiera drzwi do eksperymentowania z chatbotami, które mogą rozmawiać z ludźmi, a także tworzeniem narzędzi wspomagających naukę języków obcych.

Python może również pochwalić się tym, że jest wykorzystywany do tworzenia efektów specjalnych w wielu hollywoodzkich blockbustach. Studia filmowe takie jak Industrial Light & Magic (ILM) wykorzystują Pythona do generowania realistycznych efektów, które zachwycają

widzów na całym świecie. Dzieci, które marzą o karierze w przemyśle filmowym lub gier komputerowych, mogą być zainspirowane tym, że poprzez naukę Pythona, mogą przyczynić się do tworzenia następnej wielkiej produkcji.

Warto również wspomnieć, że Python odgrywa ważną rolę w rozwoju sztucznej inteligencji (AI) i uczenia maszynowego (ML). Biblioteki takie jak TensorFlow i Keras, które są dostępne w Pythonie, umożliwiają młodym programistom eksplorowanie tych zaawansowanych technologii. Dzieci mogą nauczyć się, jak tworzyć algorytmy, które uczą się od danych i mogą na przykład rozpoznawać obrazy, co jest niesamowicie fascynujące i praktyczne w dzisiejszym świecie technologii.

Ostatnia ciekawostka dotyczy wspólnoty wokół Pythona, która jest jedną z największych i najbardziej wspierających w świecie programowania. Społeczność ta tworzy i udostępnia bezpłatnie ogromną liczbę materiałów edukacyjnych, w tym książki, kursy online i interaktywne tutorial, które są idealne dla młodych uczniów. Dzięki temu dzieci mogą uczyć się w tempie dostosowanym do swoich indywidualnych potrzeb i mieć dostęp do wsparcia od doświadczonych programistów z całego świata.

Mając na uwadze te wszystkie aspekty, Python z pewnością stanowi atrakcyjną i inspirującą platformę dla młodych umysłów, pragnących rozpocząć swoją przygodę z kodowaniem. W miarę, jak najmłodsi programiści zanurzają się w świat Pythona, odkrywają nie tylko przyjemność z pisania kodu, ale również bogactwo możliwości, jakie przed nimi otwiera, od tworzenia gier po odkrywanie tajemnic sztucznej inteligencji.