

Michał Wiszniewski



Python na start!

Programowanie dla nastolatków

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Ewelina Burska, Michał Mrowiec

Projekt okładki: Studio Gravite / Olsztyn

Obarek, Pokoński, Pazdrijowski, Zaprucki

Grafika na okładce została wykorzystana za zgodą Shutterstock.com

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/zaprvp>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-9596-1

Copyright © Helion S.A. 2022

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

SPIS TREŚCI

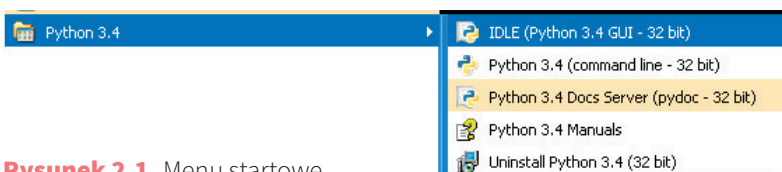
	To nie jest kolejny nudny wstęp		7
ROZDZIAŁ 1.	Instalacja potrzebnych programów		9
ROZDZIAŁ 2.	Pierwszy program, czyli jak zostałem programistą		11
ROZDZIAŁ 3.	Zmienne są niezbędne...		15
ROZDZIAŁ 4.	...a zmienne znakowe jeszcze bardziej!		19
ROZDZIAŁ 5.	Operacje wejścia — przekaż skryptowi dowolne informacje		25
ROZDZIAŁ 6.	Instrukcje warunkowe — początek prawdziwego programowania		31
ROZDZIAŁ 7.	Funkcje to jest to!		37
ROZDZIAŁ 8.	Zasięg zmiennych i nieco więcej o instrukcjach warunkowych — trzeba uważać		41
ROZDZIAŁ 9.	Pętle, czyli jak ułatwić sobie życie		47
ROZDZIAŁ 10.	Zaawansowane typy danych — to tylko groźnie brzmi		51
ROZDZIAŁ 11.	Operacje na plikach — w końcu ciekawy rozdział!		57
ROZDZIAŁ 12.	Moduły są wspaniałe!		61
ROZDZIAŁ 13.	Wyjątki, czyli błędy w języku Python		65
ROZDZIAŁ 14.	Obiekty, klasy i metody, czyli programowanie obiektowe		69
ROZDZIAŁ 15.	Przydatne funkcje i biblioteki — więcej o języku Python		79
ROZDZIAŁ 16.	Serializacja danych — zapis i odczyt danych binarnych		85
ROZDZIAŁ 17.	Pora kończyć, a tak naprawdę zaczynać		95
	Skorowidz		97

ROZDZIAŁ 2.

PIERWSZY PROGRAM, CZYLI JAK ZOSTAŁEM PROGRAMISTĄ

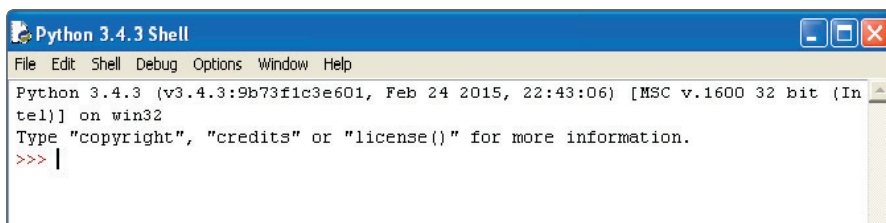
Tak jak pisałem w poprzednim rozdziale, w menu startowym pojawił się nowy wpis *Python 3.4*.

Rozwiń go i kliknij przycisk *IDLE* (ten zaznaczony na niebiesko na rysunku 2.1). Twoim oczom powinno się ukazać okno tzw. **interaktywnej powłoki**. Brzmi to poważnie, ale w rzeczywistości jest bardzo proste w użyciu.



Rysunek 2.1. Menu startowe

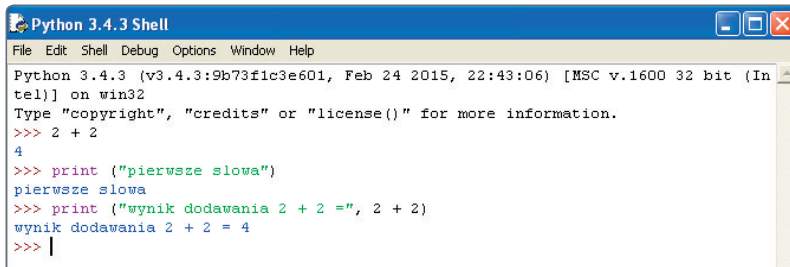
W miejscu wystąpienia znaku `>>>`, widocznego na rysunku 2.2, możesz rozpocząć wpisywanie swoich komend języka Python.



Rysunek 2.2. Powłoka języka Python

Zerknij na rysunek 2.3, przedstawiający uruchomioną powłokę.

Na początku wpisałem `2 + 2` i nacisnąłem przycisk *Enter*. Spowodowało to uruchomienie **interpretera**. Interpreter języka Python jest narzędziem, które tłumaczy to, co napisaliśmy, na język zrozumiały dla komputera. Zaraz to wyjaśnię w bardziej przystępny sposób.



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 2
4
>>> print ("pierwsze słowa")
pierwsze słowa
>>> print ("wynik dodawania 2 + 2 =", 2 + 2)
wynik dodawania 2 + 2 = 4
>>> |
```

Rysunek 2.3. Pierwszy program uruchomiony w powłoce

Pora na kilka słów o komputerach. Nie rozumieją one naszego języka i nie myślą w sposób taki jak my, dlatego aby się z nimi komunikować, potrzebujemy tłumacza, tak samo jak Węgier potrzebuje tłumacza do komunikacji z Japończykiem. Gdyby Węgier dał Japończykowi książkę napisaną w ojczystym języku Europejczyka, ten nie zrozumiałby ani słowa. Ale gdyby ten sam Węgier najpierw oddał książkę do tłumacza i przetłumaczoną wersję pokazał przyjacielowi z Azji, tamten wiedziałby dokładnie, o co chodzi Węgrowi. Tak samo jest w świecie komputerów. Nasz język i sposób rozumowania należy przetłumaczyć na język, który zrozumie komputer. W przypadku Pythona taką pracę wykonuje interpreter. Najpierw analizuje wpisaną przez nas linię (w tym przypadku `2 + 2`), a następnie tłumaczy ją na język komputera (tzw. **kod maszynowy**), który zostaje uruchomiony w kolejnym kroku. W efekcie działania `2 + 2` na ekranie pojawi się wynik, czyli 4.

Spójrz teraz na kolejną linijkę o treści:

```
print ("pierwsze słowa")
```

Słowo `print` jest komendą, która nakazuje komputerowi wyświetlić napis znajdujący się w nawiasie i w podwójnych cudzysłowach. Po naciśnięciu przycisku *Enter* pojawia się napis *pierwsze słowa*.

W każdym języku programowania występują słowa, które są odpowiedzialne za różne czynności. Jednym z takich słów jest `print`. Jest ono **funkcją**. Dzięki funkcji `print` jesteśmy w stanie wyświetlić na ekranie monitora dowolny napis. I znowu wracamy do komunikacji między człowiekiem a komputerem. Interpreter tłumaczy nasze polecenia na język komputera, ale musi wiedzieć, w jaki sposób to robić.

Świetnym przykładem, żeby to zrozumieć, jest przepis kulinarny. Wyobraźmy sobie, że interpreter jest początkującym kucharzem, który dostał polecenie przygotowania risotta. W życiu tego nie robił, więc potrzebny jest mu przepis. Sięga zatem do książki, w której znajduje się mnóstwo różnych przepisów, i z jej pomocą udaje mu się sprostać zadaniu. Słowo `print` to przepis na risotto, a książka kucharska to tzw. **biblioteka**. To znaczy, że nasz interpreter, aby przetłumaczyć słowo, sięga do biblioteki w poszukiwaniu informacji o słowie `print`. W języku Python mamy do czynienia z mnóstwem bibliotek, które ułatwiają życie programistom, a każda z nich zawiera mnóstwo „przepisów kulinarnych” na różne funkcje. Można powiedzieć, że początkowa nauka języka polega na poznawaniu najważniejszych i najczęściej używanych funkcji i to w następnych rozdziałach będziemy starali się zrobić. Prawda, że nie jest to wcale takie trudne?

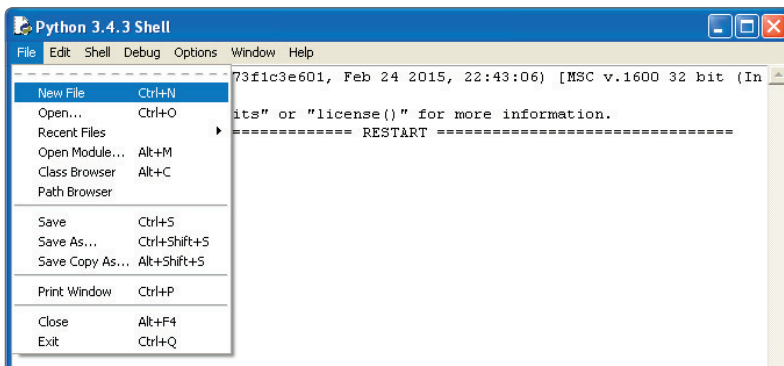
Wiedząc już, że słowo `print` jest funkcją, spójrz na następną linijkę:

```
print ("wynik dodawania 2 + 2 =", 2 + 2)
```

Funkcja `print()` pozwala na łączenie napisów z innymi elementami. Tutaj pojawia nam się napis zawarty w cudzysłowach, a następnie wykonywane jest dodawanie i jego wynik wyświetla się na ekranie. Całość w efekcie wygląda tak:

```
wynik dodawania 2 + 2 = 4
```

Żałujemy jednak, że chciałbyś napisać kilka kolejnych instrukcji, których będziesz używać co jakiś czas. Najlepszym rozwiązaniem jest zapisanie wszystkich instrukcji do pliku. W tym celu naciśnij przycisk *File*, a następnie *New File* (rysunek 2.4).



Rysunek 2.4. Menu w powłoce języka Python

Po tym zabiegu pojawi się nowe, puste okno. Wpisz w nim trzy poprzednie komendy. Wzór poniżej (listing 2.1).

LISTING 2.1. Pierwsze kroki z językiem Python

```
1 2 + 2
2 print ("pierwsze słowa")
3 print ("wynik dodawania 2 + 2 =", 2 + 2)
```

Brawo! Napisałeś swój pierwszy program, czyli tzw. **kod źródłowy**! Teraz zapisz plik, klikając *File*, a następnie *Save As...* Twój program jest gotowy do przetłumaczenia na język komputera oraz do uruchomienia. Naciśnij przycisk *F5* lub *Run*, a po nim *Run Module*. W efekcie interpreter uruchomi nowe środowisko, w którym wykona Twój program. Zauważ, że nie pojawi się tutaj wynik pierwszej linii, dlatego że nie dodałeś słowa kluczowego `print`. Poprzednim razem działanie wykonywaliśmy w otwartym środowisku i nie było to potrzebne.

Gratuluję! Przed chwilą napisałeś swój własny program, a w zasadzie skrypt. Ponieważ język Python używa interpretera, napisane w nim programy nazywamy **skryptami**. Znaczy to, że każda kolejna linia jest najpierw tłumaczona na kod maszynowy, a następnie uruchamiana, po czym interpreter „bierze się” za następną linijkę. Oznacza to, że jeżeli nasz program ma przykładowo dziesięć linii i w piątej z nich występuje jakiś błąd, to interpreter najpierw po kolei przetłumaczy i uruchomi pierwsze cztery z nich, a następnie na piątej poinformuje nas o błędzie i zakończy swoje działanie.

UWAGA

Pamiętaj, że komputer jest brutalnie konsekwentny. Oznacza to, że każdy znaczek, kropka, litera czy cyfra ma dla niego znaczenie, dlatego pisząc swój program, upewnij się, że nie ma w nim niepotrzebnych błędów.

ZADANIA

1. Utwórz nowy plik źródłowy. Użyj w nim funkcji `print` do wyświetlenia wyników dodawania $3 + 3$ oraz $7 + 8$. Wynik powinien wyglądać następująco:

```
Wynik dodawania 3 + 3 = 6, a 7 + 8 = 15.
```

2. Spróbuj uruchomić powyższy skrypt bezpośrednio w środowisku IDLE.

SKOROWIDZ

A

adres pamięci, 20
analiza zasięgu zmiennych, 41
argumenty funkcji, 38

B

biblioteka, 13
 datetime, 81
 math, 80
 os, 81
 random, 79
 smtp, 82
 sys, 28
 tkinter, 63
 winsound, 82
błąd składni, 65

C

cechy zmiennych znakowych, 21
cmd, 28

D

dane binarne, 85
data i czas, 81
destruktor, 71
dostęp do systemu operacyjnego, 81
dziecko, 75
dziedziczenie, 74
dźwięk, 82

E

edytowanie zmiennej systemowej, 27
elementy biblioteki tkinter, 63
e-mail, 82

F

funkcja, 37
 capitalize(), 23
 close(), 58
 cmp(), 53
 exit(), 35
 input(), 25
 join(), 82
 len(), 53
 open(), 58
 print(), 13
 tuple(), 53

G

generowanie losowych danych, 79
gra, 48

I

instalacja oprogramowania, 9
instrukcja warunkowa if, 31, 33
interaktywna powłoka, 11
interpreter, 12

K

klasa, 69
kod
 maszynowy, 12
 źródłowy, 14
komenda
 cmd, 28
 cd, 28
konsola programu sqlite3, 90
konstruktor, 71
krotka, 53

L

lista, 51

M

menu

- startowe, 11
- w powłoce, 13

metoda, 52, 69

- append(), 52
- extend(), 52
- destroy(), 63
- maniloop(), 63
- ctime(), 81

moduł, 23, 61

O

obiekt, 69

obsługa błędów, 33, 66

odczyt danych binarnych, 85

odtworzenie dźwięku, 82

okno linii komend, 28

operacje

- matematyczne, 80
- na plikach, 57
- wejścia, 25

operator

- and, 45
- del, 73
- modulo, 49
- or, 45

operatory arytmetyczne, 17

oprogramowanie, 9

P

pamięć operacyjna, 15, 20

pętla

- for, 48
- while, 47

pierwszy program, 11

pliki, 57

- binarne, 59
- tekstowe, 57

powłoka języka, 11

program sqlite3, 90

programowanie obiektowe, 69

przeciążanie metod, 76

przekazywanie argumentów, 28

R

rodzic, 75

rzutowanie, 35

S

serializacja danych, 85

skrypt, 7, 14

słownik, 54

słowo kluczowe

- and, 43
- continue, 48
- del, 53
- exception, 66
- global, 42
- int, 26
- or, 43
- return, 38

struktury sekwencyjne, 54

T

tworzenie

klas, 69

wyjątków, 66

zmiennych, 15

typy danych, 51

float, 16

string, 22

int, 26

U

uchwyt do pliku, 57

ukrywanie metod, 74

uruchamianie linii komend, 28

W

wiadomości e-mail, 82

wprowadzanie danych, 25

wyciek pamięci, 15, 73

wyjątek, 65, 67

typu IOException, 66

Z

zaawansowane

operacje matematyczne, 80

typy danych, 51

zapis danych binarnych, 85

zasięg zmiennych, 41

zmienna, 15

systemowa PATH, 28

zmienne

środowiskowe, 27

globalne, 42

lokalne, 41

znakowe, 19, 34

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

Czy kiedykolwiek zastanawiałeś się nad tym, dlaczego na świecie istnieje aż tyle języków programowania? Czym się od siebie różnią? I których z nich warto się nauczyć? Na pewno co najmniej kilku, ale na Twojej liście nie może zabraknąć Pythona! Ten język ma dwie potężne zalety: po pierwsze, jest znacznie łatwiejszy w nauce i bardziej intuicyjny niż choćby C czy C++, a po drugie, umożliwia tworzenie skryptów, czyli funkcjonalnych programów, przydających się na co dzień programistom wszelkiej maści, administratorom sieci i wielu innym ludziom. Python to także fantastyczne narzędzie do tworzenia rozbudowanych projektów. Zresztą możesz się o tym szybko przekonać.

W tej książce znajdziesz omówienie wszystkich najważniejszych funkcji i możliwości Pythona. Dowiesz się, jak pracować w środowisku tego języka, jak używać zmiennych, instrukcji warunkowych, funkcji i typów danych. Zobaczysz, w jakich sytuacjach warto wykorzystywać pętle i moduły, jak wykonywać operacje na plikach i jak obsługiwać wyjątki. Sprawdzisz, do czego przydaje się programowanie strukturalne, a do czego obiektowe, i odkryjesz, jak łatwo połączyć je w Pythonie! Krótko mówiąc, zanim się obejrzyś, zaczniesz samodzielnie pisać zaawansowane skrypty. Sprawdź, programuj i baw się dobrze!

- Pierwszy program, czyli jak zostałem programistą
- Zmienne są niezbędne... a zmienne znakowe jeszcze bardziej
- Operacje wejścia — przekaz skryptowi dowolne informacje
- Instrukcje warunkowe — początek prawdziwego programowania
- Funkcje — to jest to!
- Zasięg zmiennych i nieco więcej o instrukcjach warunkowych — trzeba uważać
- Pętle, czyli jak ułatwić sobie życie
- Zaawansowane typy danych — to tylko groźnie brzmi!
- Operacje na plikach — w końcu ciekawy rozdział!
- Moduły — one są wspaniałe!
- Wyjątki, czyli błędy w języku Python
- Obiekty, klasy i metody, czyli programowanie obiektowe

Zaklinaj węża
— programuj w Pythonie!

  helion.pl	<i>Sprawdź nasze szkolenia!</i>  AKADEMIA IT & BUSINESS HELIONSZKOLENIA.PL	KOD KORZYŚCI <i>Sięgnij po więcej!</i>   ISBN 978-83-283-9596-1  9 788328 395961
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 39,00 zł