

Spis treści

O AUTORZE	23
O KOREKTORZE MERYTORYCZNYM	23
PODZIĘKOWANIA	25
WPROWADZENIE DO DRUGIEGO WYDANIA KSIĄŻKI	27
WPROWADZENIE	31
Do kogo jest skierowana ta książka?	32
Czego nauczysz się z tej książki?	32
Zasoby w internecie	34
Dlaczego Python?	34

CZĘŚĆ I. PODSTAWY

I

ROZPOCZĘCIE PRACY	39
Przygotowanie środowiska programistycznego	39
Wersje Pythona	39
Wykonanie fragmentu kodu w Pythonie	40
Edytor tekstu Sublime Text	41
Python w różnych systemach operacyjnych	41
Python w systemie Windows	41
Python w systemie macOS	43
Python w systemach z rodziny Linux	45

Uruchomienie programu typu „Witaj, świecie!”	46
Konfiguracja Sublime Text dla Pythona 3	46
Uruchomienie programu typu „Witaj, świecie!”	47
Rozwiązywanie problemów podczas instalacji	47
Uruchamianie programów Pythona z poziomu powłoki	49
W systemie Windows	49
W systemach macOS i Linux	49
Podsumowanie	51
 2	
ZMIENNE I PROSTE TYPY DANYCH	53
Co tak naprawdę dzieje się po uruchomieniu <code>hello_world.py</code> ?	53
Zmienne	54
Nadawanie nazw zmiennym i używanie zmiennych	55
Unikanie błędów związanych z nazwami podczas używania zmiennych	56
Zmienna to etykieta	57
Ciągi tekstowe	58
Zmiana wielkości liter ciągu tekstowego za pomocą metod	58
Używanie zmiennych w ciągach tekstowych	60
Dodawanie białych znaków do ciągów tekstowych	
za pomocą tabulatora i znaku nowego wiersza	61
Usunięcie białych znaków	62
Unikanie błędów składni w ciągach tekstowych	63
Liczby	65
Liczby całkowite	66
Liczby zmiennoprzecinkowe	66
Liczby całkowite i zmiennoprzecinkowe	67
Znaki podkreślenia w liczbach	68
Wiele przypisań	68
Stałe	68
Komentarze	69
Jak można utworzyć komentarz?	69
Jakiego rodzaju komentarze należy tworzyć?	70
Zen Pythona	70
Podsumowanie	72
 3	
WPROWADZENIE DO LIST	73
Czym jest lista?	73
Uzyskanie dostępu do elementów listy	74
Numeracja indeksu zaczyna się od 0, a nie od 1	75
Użycie poszczególnych wartości listy	75

Zmienianie, dodawanie i usuwanie elementów	76
Modyfikowanie elementów na liście	76
Dodawanie elementów do listy	77
Usuwanie elementu z listy	79
Organizacja listy	84
Trwałe sortowanie listy za pomocą metody sort()	84
Tymczasowe sortowanie listy za pomocą funkcji sorted()	85
Wyświetlanie listy w odwrotnej kolejności alfabetycznej	86
Określenie wielkości listy	86
Unikanie błędów indeksu podczas pracy z listą	88
Podsumowanie	89
4	
PRACA Z LISTĄ	91
Iteracja przez całą listę	91
Dokładniejsza analiza pętli	92
Wykonanie większej liczby zadań w pętli for	93
Wykonywanie operacji po pętli for	95
Unikanie błędów związanych z wcięciami	96
Brak wcięcia	96
Brak wcięcia dodatkowych wierszy	97
Niepotrzebne wcięcie	97
Niepotrzebne wcięcie po pętli	98
Brak dwukropka	99
Tworzenie list liczbowych	100
Użycie funkcji range()	100
Użycie funkcji range() do utworzenia listy liczb	101
Proste dane statystyczne dotyczące listy liczb	103
Lista składana	103
Praca z fragmentami listy	105
Wycinek listy	105
Iteracja przez wycinek	107
Kopiowanie listy	107
Krotka	110
Definiowanie krotki	111
Iteracja przez wszystkie wartości krotki	112
Nadpisanie krotki	112
Styl tworzonego kodu	113
Konwencje stylu	114
Wcięcia	114
Długość wiersza	114
Puste wiersze	115
Inne specyfikacje stylu	115
Podsumowanie	116

5	KONSTRUKCJA IF	117
	Prosty przykład	117
	Test warunkowy	118
	Sprawdzenie równości	118
	Ignorowanie wielkości liter podczas sprawdzania równości	119
	Sprawdzenie nierówności	120
	Porównania liczbowe	121
	Sprawdzanie wielu warunków	122
	Sprawdzanie, czy wartość znajduje się na liście	123
	Sprawdzanie, czy wartość nie znajduje się na liście	124
	Wyrażenie boolowskie	124
	Polecenie if	125
	Proste polecenia if	125
	Polecenia if-else	127
	Łańcuch if-elif-else	127
	Użycie wielu bloków elif	129
	Pominięcie bloku else	130
	Sprawdzanie wielu warunków	130
	Używanie poleceń if z listami	134
	Sprawdzanie pod kątem wartości specjalnych	134
	Sprawdzanie, czy lista nie jest pusta	135
	Użycie wielu list	136
	Nadawanie stylu poleceniom if	138
	Podsumowanie	139
6	SŁOWNIKI	141
	Prosty słownik	142
	Praca ze słownikami	142
	Uzyskiwanie dostępu do wartości słownika	143
	Dodanie nowej pary klucz-wartość	144
	Rozpoczęcie pracy od pustego słownika	145
	Modyfikowanie wartości słownika	145
	Usuwanie pary klucz-wartość	147
	Słownik podobnych obiektów	147
	Używanie metody get() w celu uzyskania dostępu do wartości	149
	Iteracja przez słownik	151
	Iteracja przez wszystkie pary klucz-wartość	151
	Iteracja przez wszystkie klucze słownika	153
	Iteracja przez uporządkowane klucze słownika	155
	Iteracja przez wszystkie wartości słownika	156

Zagnieżdżanie	158
Lista słowników	158
Lista w słowniku	161
Słownik w słowniku	164
Podsumowanie	166

7

DANE WEJŚCIOWE UŻYTKOWNIKA I PĘTLA WHILE 167

Jak działa funkcja input()?	168
Przygotowanie jasnych i zrozumiałych komunikatów	169
Użycie funkcji int() do akceptowania liczbowych danych wejściowych	170
Operator modulo	171
Wprowadzenie do pętli while	173
Pętla while w działaniu	173
Umożliwienie użytkownikowi podjęcia decyzji o zakończeniu działania programu	174
Użycie flagi	175
Użycie polecenia break do opuszczenia pętli	177
Użycie polecenia continue w pętli	178
Unikanie pętli działającej w nieskończoność	178
Użycie pętli while wraz z listami i słownikami	180
Przenoszenie elementów z jednej listy na drugą	180
Usuwanie z listy wszystkich egzemplarzy określonej wartości	182
Umieszczenie w słowniku danych wejściowych wprowadzonych przez użytkownika	182
Podsumowanie	184

8

FUNKCJE 185

Definiowanie funkcji	185
Przekazywanie informacji do funkcji	186
Argumenty i parametry	187
Przekazywanie argumentów	188
Argumenty pozycyjne	188
Argumenty w postaci słów kluczowych	190
Wartości domyślne	191
Odpowiedniki wywołań funkcji	192
Unikanie błędów związanych z argumentami	193
Wartość zwrotna	194
Zwrot prostej wartości	195
Definiowanie argumentu jako opcjonalnego	196
Zwrot słownika	197
Używanie funkcji wraz z pętlą while	198

Przekazywanie listy	201
Modyfikowanie listy w funkcji	201
Uniemożliwianie modyfikowania listy przez funkcję	204
Przekazywanie dowolnej liczby argumentów	205
Argumenty pozycyjne i przekazywanie dowolnej liczby argumentów	207
Używanie dowolnej liczby argumentów w postaci słów kluczowych	208
Przechowywanie funkcji w modułach	210
Import całego modułu	210
Import określonych funkcji	211
Użycie słowa kluczowego <code>as</code> w celu zdefiniowania aliasu funkcji	212
Użycie słowa kluczowego <code>as</code> w celu zdefiniowania aliasu modułu	212
Import wszystkich funkcji modułu	213
Nadawanie stylu funkcjom	214
Podsumowanie	215
9	
KLASY	217
Utworzenie i użycie klasy	218
Utworzenie klasy <code>Dog</code>	218
Utworzenie egzemplarza na podstawie klasy	220
Praca z klasami i egzemplarzami	223
Klasa <code>Car</code>	223
Przypisanie atrybutowi wartości domyślnej	224
Modyfikacja wartości atrybutu	225
Dziedziczenie	229
Metoda <code>__init__()</code> w klasie potomnej	229
Definiowanie atrybutów i metod dla klasy potomnej	231
Nadpisywanie metod klasy nadrzędnej	232
Egzemplarz jako atrybut	233
Modelowanie rzeczywistych obiektów	236
Import klas	236
Import pojedynczej klasy	236
Przechowywanie wielu klas w module	239
Import wielu klas z modułu	240
Import całego modułu	241
Import wszystkich klas z modułu	242
Import modułu w module	242
Używanie aliasów	243
Określenie swojego sposobu pracy	244
Biblioteka standardowa Pythona	245
Nadawanie stylu klasom	245
Podsumowanie	247

PLIKI I WYJĄTKI	249
Odczytywanie danych z pliku	250
Wczytywanie całego pliku	250
Ścieżka dostępu do pliku	252
Odczytywanie wiersz po wierszu	253
Utworzenie listy wierszy na podstawie zawartości pliku	255
Praca z zawartością pliku	255
Ogromne pliki, czyli na przykład milion cyfr	256
Czy data Twoich urodzin znajduje się w liczbie pi?	257
Zapisywanie danych w pliku	259
Zapisywanie danych do pustego pliku	259
Zapisywanie wielu wierszy	260
Dołączanie do pliku	261
Wyjątki	262
Obsługiwanie wyjątku ZeroDivisionError	262
Używanie bloku try-except	263
Używanie wyjątków w celu uniknięcia awarii programu	263
Blok else	264
Obsługa wyjątku FileNotFoundError	266
Analiza tekstu	267
Praca z wieloma plikami	268
Ciche niepowodzenie	270
Które błędy należy zgłaszać?	271
Przechowywanie danych	272
Używanie json.dump() i json.load()	273
Zapisywanie i odczytywanie danych wygenerowanych przez użytkownika	274
Refaktoryzacja	276
Podsumowanie	279
 II	
TESTOWANIE KODU	281
Testowanie funkcji	282
Test jednostkowy i zestaw testów	283
Zaliczenie testu	283
Niezaliczenie testu	285
Reakcja na niezaliczony test	287
Dodanie nowego testu	288
Testowanie klasy	290
Różne rodzaje metod asercji	290
Klasa do przetestowania	290
Testowanie klasy AnonymousSurvey	293
Metoda setUp()	295
Podsumowanie	297

CZĘŚĆ II. PROJEKTY

PROJEKT I. INWAZJA OBCYCH 301

12

STATEK, KTÓRY STRZELA POCISKAMI 303

Planowanie projektu	304
Instalacja Pygame	304
Rozpoczęcie pracy nad projektem gry	305
Utworzenie okna Pygame i reagowanie na działania użytkownika	305
Zdefiniowanie koloru tła	307
Utworzenie klasy ustawień	308
Dodanie obrazu statku kosmicznego	309
Utworzenie klasy statku kosmicznego	310
Wyświetlenie statku kosmicznego na ekranie	312
Refaktoryzacja, czyli metody <code>_check_events()</code> i <code>_update_screen()</code>	314
Metoda <code>_check_events()</code>	314
Metoda <code>_update_screen()</code>	315
Kierowanie statkiem kosmicznym	316
Reakcja na naciśnięcie klawisza	316
Umożliwienie nieustannego ruchu	316
Poruszanie statkiem w obu kierunkach	318
Dostosowanie szybkości statku	320
Ograniczenie zasięgu poruszania się statku	322
Refaktoryzacja metody <code>_check_events()</code>	323
Naciśnięcie klawisza Q w celu zakończenia gry	323
Uruchamianie gry w trybie pełnoekranowym	324
Krótkie powtórzenie	325
alien_invasion.py	325
settings.py	325
ship.py	325
Wystrzeliwanie pocisków	326
Dodawanie ustawień dotyczących pocisków	326
Utworzenie klasy Bullet	327
Przechowywanie pocisków w grupie	328
Wystrzeliwanie pocisków	329
Usuwanie niewidocznych pocisków	330
Ograniczenie liczby pocisków	332
Utworzenie metody <code>_update_bullets()</code>	333
Podsumowanie	334

13

OBCY!	335
Przegląd projektu	336
Utworzenie pierwszego obcego	336
Utworzenie klasy Alien	337
Utworzenie egzemplarza obcego	338
Utworzenie floty obcych	339
Ustalenie maksymalnej liczby obcych wyświetlanych w jednym rzędzie	340
Utworzenie rzędów obcych	341
Refaktoryzacja metody <code>_create_fleet()</code>	342
Dodawanie rzędów	343
Poruszanie flotą obcych	345
Przesunięcie obcych w prawo	346
Zdefiniowanie ustawień dla kierunku poruszania się floty	347
Sprawdzenie, czy obcy dotarł do krawędzi ekranu	348
Przesunięcie floty w dół i zmiana kierunku	349
Zestrzeliwanie obcych	350
Wykrywanie kolizji z pociskiem	350
Utworzenie większych pocisków w celach testowych	351
Ponowne utworzenie floty	353
Zwiększenie szybkości pocisku	353
Refaktoryzacja metody <code>_update_bullets()</code>	354
Zakończenie gry	355
Wykrywanie kolizji między obcym i statkiem	355
Reakcja na kolizję między obcym i statkiem	356
Obcy, który dociera do dolnej krawędzi ekranu	359
Koniec gry!	360
Ustalenie, które komponenty gry powinny być uruchomione	361
Podsumowanie	362

14

PUNKTACJA	363
Dodanie przycisku Gra	363
Utworzenie klasy Button	364
Wyświetlenie przycisku na ekranie	366
Uruchomienie gry	367
Zerowanie gry	368
Dezaktywacja przycisku Gra	369
Ukrycie kursora myszy	369
Zmiana poziomu trudności	371
Zmiana ustawień dotyczących szybkości	371
Wyzerowanie szybkości	373

Punkcja	373
Wyświetlanie punkcji	374
Utworzenie tablicy wyników	375
Uaktualnienie punkcji po zestrzeleniu obcego	377
Zerowanie wyniku	378
Zagwarantowanie uwzględnienia wszystkich trafień	378
Zwiększenie liczby zdobywanych punktów	379
Zaokrąglenie punkcji	380
Najlepsze wyniki	381
Wyświetlenie aktualnego poziomu gry	384
Wyświetlenie liczby statków	387
Podsumowanie	390
PROJEKT 2. WIZUALIZACJA DANYCH	391
15	
GENEROWANIE DANYCH	393
Instalacja matplotlib	394
Wygenerowanie prostego wykresu liniowego	395
Zmienianie etykiety i grubości wykresu	395
Poprawianie wykresu	397
Używanie wbudowanych stylów	398
Używanie funkcji scatter() do wyświetlania poszczególnych punktów i nadawania im stylu	400
Wyświetlanie serii punktów za pomocą funkcji scatter()	401
Automatyczne obliczanie danych	402
Definiowanie własnych kolorów	403
Użycie mapy kolorów	404
Automatyczny zapis wykresu	404
Błądzenie losowe	405
Utworzenie klasy RandomWalk	406
Wybór kierunku	407
Wyświetlenie wykresu błędzenia losowego	408
Wygenerowanie wielu błędzeń losowych	408
Nadawanie stylu danym wygenerowanym przez błędzenie losowe	410
Symulacja rzutu kością do gry za pomocą plotly	415
Instalacja plotly	416
Utworzenie klasy Die	416
Rzut kością do gry	417
Analiza wyników	417
Utworzenie histogramu	418
Rzut dwiema kośćmi	420
Rzut kośćmi o różnej liczbie ścianek	422
Podsumowanie	424

POBIERANIE DANYCH 425

Format CSV	426
Przetwarzanie nagłówków pliku CSV	426
Wyświetlanie nagłówków i ich położenia	427
Wyodrębnienie i odczytanie danych	428
Wyświetlenie danych na wykresie temperatury	429
Moduł datetime	430
Wyświetlanie daty	431
Wyświetlenie dłuższego przedziału czasu	432
Wyświetlenie drugiej serii danych	433
Nakładanie cienia na wykresie	435
Sprawdzenie pod kątem błędów	436
Samodzielne pobieranie danych	440
Mapowanie globalnych zbiorów danych — format JSON	441
Pobranie danych dotyczących trzęsień ziemi	442
Analiza danych JSON	442
Utworzenie listy trzęsień ziemi	445
Wyodrębnienie siły trzęsienia ziemi	445
Wyodrębnienie danych o miejscu wystąpienia trzęsienia ziemi	446
Budowanie mapy świata	447
Inny sposób określenia danych wykresu	448
Dostosowanie wielkości punktu	449
Dostosowanie koloru punktu	450
Inne skale kolorów	452
Dodanie tekstu wyświetlanego po wskazaniu punktu na mapie	452
Podsumowanie	454

PRACA Z API 455

Użycie Web API	455
Git i GitHub	456
Żądanie danych za pomocą wywołania API	456
Instalacja requests	457
Przetworzenie odpowiedzi API	458
Praca ze słownikiem odpowiedzi	459
Podsumowanie repozytoriów najczęściej oznaczanych gwiazdką	461
Monitorowanie ograniczeń liczby wywołań API	463
Wizualizacja repozytoriów za pomocą pakietu plotly	463
Dopracowanie wykresów generowanych przez plotly	465
Dodanie własnych odpowiedzi	467
Dodawanie łączy do wykresu	469
Więcej o plotly i API GitHub	470
Hacker News API	470
Podsumowanie	475



PROJEKT 3. APLIKACJE INTERNETOWE 477

18

ROZPOCZĘCIE PRACY Z DJANGO 479

Przygotowanie projektu	480
Opracowanie specyfikacji	480
Utworzenie środowiska wirtualnego	480
Aktywacja środowiska wirtualnego	481
Instalacja frameworka Django	481
Utworzenie projektu w Django	482
Utworzenie bazy danych	483
Przegląd projektu	484
Uruchomienie aplikacji	485
Definiowanie modeli	486
Aktywacja modeli	487
Witryna administracyjna Django	489
Zdefiniowanie modelu Entry	492
Migracja modelu Entry	493
Rejestracja modelu Entry w witrynie administracyjnej	494
Powłoka Django	495
Tworzenie stron internetowych — strona główna aplikacji	496
Mapowanie adresu URL	498
Utworzenie widoku	500
Utworzenie szablonu	500
Utworzenie dodatkowych stron	502
Dziedziczenie szablonu	502
Strona tematów	505
Strony poszczególnych tematów	508
Podsumowanie	512

19

KONTA UŻYTKOWNIKÓW 513

Umożliwienie użytkownikom wprowadzania danych	514
Dodawanie nowego tematu	514
Dodawanie nowych wpisów	518
Edycja wpisu	523
Konfiguracja kont użytkowników	527
Aplikacja users	527
Strona logowania	528
Wylogowanie	531
Strona rejestracji użytkownika	532
Umożliwienie użytkownikom bycia właścicielami swoich danych	536
Ograniczenie dostępu za pomocą dekoratora @login_required	536
Powiązanie danych z określonymi użytkownikami	538
Przyznanie dostępu jedynie odpowiednim użytkownikom	541

Ochrona tematów użytkownika	542
Ochrona strony edit_entry	543
Powiązanie nowego tematu z bieżącym użytkownikiem	544
Podsumowanie	545

20

NADANIE STYLU I WDROŻENIE APLIKACJI 547

Nadanie stylu aplikacji Learning Log	548
Aplikacja django-bootstrap4	548
Użycie Bootstrapa do nadania stylu aplikacji Learning Log	549
Modyfikacja pliku base.html	549
Użycie elementu Jumbotron do nadania stylu stronie głównej	554
Nadanie stylu stronie logowania	556
Nadanie stylu stronie tematów	557
Nadanie stylów wpisom na stronie tematu	558
Wdrożenie aplikacji Learning Log	560
Utworzenie konta w Heroku	560
Instalacja Heroku CLI	560
Instalacja wymaganych pakietów	560
Utworzenie listy pakietów w pliku requirements.txt	561
Określenie środowiska uruchomieniowego Pythona	562
Modyfikacja pliku settings.py dla Heroku	562
Utworzenie pliku Procfile do uruchomienia procesu	563
Użycie Gita do monitorowania plików projektu	563
Przekazanie projektu do Heroku	566
Konfiguracja bazy danych w Heroku	567
Dopracowanie wdrożenia projektu w Heroku	568
Zabezpieczenie wdrożonego projektu	570
Zatwierdzenie zmian i przekazanie ich do serwera	571
Zdefiniowanie zmiennej środowiskowej w Heroku	572
Utworzenie własnych stron błędu	573
Nieustanna rozbudowa	576
Opcja SECRET_KEY	577
Usunięcie projektu z Heroku	577
Podsumowanie	578

POSŁOWIE 581

A

INSTALACJA PYTHONA I ROZWIĄZYWANIE PROBLEMÓW 583

Python w Windows	583
Odszukanie interpretera Pythona	584
Dodanie Pythona do zmiennej Path	584
Ponowna instalacja Pythona	585

Python w systemie macOS	585
Instalacja Homebrew	586
Instalacja Pythona	586
Python w systemie Linux	587
Słowa kluczowe Pythona i wbudowane funkcje	587
Słowa kluczowe Pythona	588
Wbudowane funkcje Pythona	588

B

EDYTORY TEKSTU I ŚRODOWISKA IDE 589

Dostosowanie ustawień edytora Sublime Text	590
Konwersja tabulatorów na spacje	590
Ustawianie wskaźnika długości linii	590
Wcięcia i brak wcięć bloków kodu	591
Umieszczenie bloku kodu w komentarzu	591
Zapisywanie konfiguracji edytora Sublime Text	591
Dalsze konfigurowanie edytora Sublime Text	592
Inne edytory tekstu i środowiska IDE	592
IDLE	592
Geany	592
Emacs i vim	592
Atom	593
Visual Studio Code	593
PyCharm	593

C

UZYSKIWANIE POMOCY 595

Pierwsze kroki	595
Spróbuj jeszcze raz	596
Chwila odpoczynku	596
Korzystaj z zasobów tej książki	597
Wyszukiwanie informacji w internecie	597
Stack Overflow	597
Oficjalna dokumentacja Pythona	598
Oficjalna dokumentacja biblioteki	598
r/learnpython	598
Posty na blogach	598
Kanały IRC	599
Założenie konta na kanale IRC	599
Kanały, do których warto się przyłączyć	599
Kultura na kanale IRC	600
Slack	600
Discord	600

D**UŻYWANIE GITA DO KONTROLI WERSJI 603**

Instalacja Gita	604
Instalacja Gita w systemie Windows	604
Instalacja Gita w systemie macOS	604
Instalacja Gita w systemie Linux	604
Konfiguracja Gita	604
Tworzenie projektu	605
Ignorowanie plików	605
Inicjalizacja repozytorium	606
Sprawdzanie stanu	606
Dodawanie plików do repozytorium	607
Zatwierdzanie plików	607
Sprawdzanie dziennika projektu	608
Drugie zatwierdzenie	609
Przywracanie stanu projektu	610
Przywrócenie projektu do wcześniejszego stanu	611
Usunięcie repozytorium	613

wania typu open source. W czasie wolnym współpracuje po godzinach z innymi osobami z branży.

© korektorze
merytorycznym

Kenneth Love jest już od wielu lat nauczycielem programisty Pythona i organizatorem konferencji. Wielokrotnie występowal na konferencjach, przeprowadzał profesjonalne szkolenia i realizował różne projekty Pythona i Django. Obecnie jest inżynierem oprogramowania w O'Reilly Media. Kenneth jest również współautorem książki Django-recipes, który dostarcza wiele użytecznych odpowiedzi dla użytkowników opartych na klasach we frameworku Django. Możesz się z nim skontaktować za pomocą serwisu Twitter, w którym znajdziesz go pod nickiem @kenanrlama.