

Spis treści

WPROWADZENIE DO TRZECIEGO WYDANIA KSIĄŻKI	21
PODZIĘKOWANIA	24
WPROWADZENIE	26
CZĘŚĆ I. PODSTAWY	31
1	
ROZPOCZĘCIE PRACY	33
Przygotowanie środowiska programistycznego	33
Wersje Pythona	33
Wykonanie fragmentu kodu w Pythonie	34
Edytor tekstu VS Code	34
Python w różnych systemach operacyjnych	35
Python w systemie Windows	35
Python w systemie macOS	37
Python w systemach z rodziny Linux	39
Uruchomienie programu typu „Witaj, świecie!”	40
Instalacja w VS Code rozszerzenia przeznaczonego do obsługi Pythona	40
Uruchomienie programu typu „Witaj, świecie!”	41
Rozwiązywanie problemów podczas instalacji	42
Uruchamianie programów Pythona z poziomu powłoki	43
W systemie Windows	43
W systemach macOS i Linux	44
Podsumowanie	45
2	
ZMIENNE I PROSTE TYPY DANYCH	46
Co tak naprawdę dzieje się po uruchomieniu hello_world.py?	46
Zmienne	47
Nadawanie nazw zmiennym i używanie zmiennych	48
Unikanie błędów związanych z nazwami podczas używania zmiennych	49
Zmienna to etykieta	50

Ciągi tekstowe	51
Zmiana wielkości liter ciągu tekstowego za pomocą metod	51
Używanie zmiennych w ciągach tekstowych	53
Dodawanie białych znaków do ciągów tekstowych za pomocą tabulatora i znaku nowego wiersza	54
Usunięcie białych znaków	55
Usunięcie prefiksu	56
Unikanie błędów składni w ciągach tekstowych	57
Liczby	59
Liczby całkowite	59
Liczby zmiennoprzecinkowe	60
Liczby całkowite i zmiennoprzecinkowe	61
Znaki podkreślenia w liczbach	61
Wiele przypisań	62
Stałe	62
Komentarze	63
Jak można utworzyć komentarz?	63
Jakiego rodzaju komentarze należy tworzyć?	63
Zen Pythona	64
Podsumowanie	66
3	
WPROWADZENIE DO LIST	67
Czym jest lista?	67
Uzyskanie dostępu do elementów listy	68
Numeracja indeksu zaczyna się od 0, a nie od 1	68
Użycie poszczególnych wartości listy	69
Zmianie, dodawanie i usuwanie elementów	70
Modyfikowanie elementów na liście	70
Dodawanie elementów do listy	71
Usuwanie elementu z listy	73
Organizacja listy	77
Trwałe sortowanie listy za pomocą metody sort()	78
Tymczasowe sortowanie listy za pomocą funkcji sorted()	78
Wyświetlanie listy w odwrotnej kolejności alfabetycznej	80
Określenie wielkości listy	80
Unikanie błędów indeksu podczas pracy z listą	81
Podsumowanie	83
4	
PRACA Z LISTĄ	84
Iteracja przez całą listę	84
Dokładniejsza analiza pętli	85
Wykonanie większej liczby zadań w pętli for	86
Wykonywanie operacji po pętli for	88

Unikanie błędów związanych z wcięciami	89
Brak wcięcia	89
Brak wcięcia dodatkowych wierszy	90
Niepotrzebne wcięcie	90
Niepotrzebne wcięcie po pętli	91
Brak dwukropka	92
Tworzenie list liczbowych	93
Użycie funkcji range()	93
Użycie funkcji range() do utworzenia listy liczb	94
Proste dane statystyczne dotyczące listy liczb	96
Lista składana	96
Praca z fragmentami listy	98
Wycinek listy	98
Iteracja przez wycinek	100
Kopiowanie listy	101
Krotka	104
Definiowanie krotki	104
Iteracja przez wszystkie wartości krotki	105
Nadpisanie krotki	105
Styl tworzonego kodu	106
Konwencje stylu	107
Wcięcia	107
Długość wiersza	108
Puste wiersze	108
Inne specyfikacje stylu	108
Podsumowanie	109
5	
KONSTRUKCJA IF	110
Prosty przykład	110
Test warunkowy	111
Sprawdzenie równości	111
Ignorowanie wielkości liter podczas sprawdzania równości	112
Sprawdzenie nierówności	113
Porównania liczbowe	114
Sprawdzanie wielu warunków	115
Sprawdzanie, czy wartość znajduje się na liście	116
Sprawdzanie, czy wartość nie znajduje się na liście	116
Wyrażenie boolowskie	117
Polecenie if	118
Proste polecenia if	118
Polecenia if-else	119
Konstrukcja if-elif-else	120
Użycie wielu bloków elif	122
Pominięcie bloku else	123
Sprawdzanie wielu warunków	123

Używanie poleceń if z listami	126
Sprawdzanie pod kątem wartości specjalnych	126
Sprawdzanie, czy lista nie jest pusta	128
Użycie wielu list	129
Nadawanie stylu poleceniom if	131
Podsumowanie	132

6

SŁOWNIKI

Prosty słownik	133
Praca ze słownikami	134
Uzyskiwanie dostępu do wartości słownika	135
Dodanie nowej pary klucz-wartość	136
Rozpoczęcie pracy od pustego słownika	137
Modyfikowanie wartości słownika	137
Usuwanie pary klucz-wartość	139
Słownik podobnych obiektów	139
Używanie metody get() w celu uzyskania dostępu do wartości	141
Iteracja przez słownik	143
Iteracja przez wszystkie pary klucz-wartość	143
Iteracja przez wszystkie klucze słownika	145
Iteracja przez uporządkowane klucze słownika	147
Iteracja przez wszystkie wartości słownika	148
Zagnieżdżanie	150
Lista słowników	150
Lista w słowniku	153
Słownik w słowniku	156
Podsumowanie	158

7

DANE WEJŚCIOWE UŻYTKOWNIKA I PĘTLA WHILE

Jak działa funkcja input()?	160
Przygotowanie jasnych i zrozumiałych komunikatów	160
Użycie funkcji int() do akceptowania liczbowych danych wejściowych	162
Operator modulo	163
Wprowadzenie do pętli while	164
Pętla while w działaniu	165
Umożliwienie użytkownikowi podjęcia decyzji o zakończeniu działania programu	165
Użycie flagi	167
Użycie polecenia break do opuszczenia pętli	169
Użycie polecenia continue w pętli	170
Unikanie pętli działającej w nieskończoność	170

Użycie pętli while wraz z listami i słownikami	172
Przenoszenie elementów z jednej listy na drugą	172
Usuwanie z listy wszystkich egzemplarzy określonej wartości	174
Umieszczenie w słowniku danych wejściowych wprowadzonych przez użytkownika	174
Podsumowanie	176

8

FUNKCJE	177
Definiowanie funkcji	177
Przekazywanie informacji do funkcji	178
Argumenty i parametry	179
Przekazywanie argumentów	180
Argumenty pozycyjne	180
Argumenty w postaci słów kluczowych	182
Wartości domyślne	183
Odpowiedniki wywołań funkcji	184
Unikanie błędów związanych z argumentami	185
Wartość zwrotna	186
Zwrot prostej wartości	187
Definiowanie argumentu jako opcjonalnego	187
Zwrot słownika	189
Używanie funkcji wraz z pętlą while	190
Przekazywanie listy	193
Modyfikowanie listy w funkcji	193
Uniemożliwianie modyfikowania listy przez funkcję	196
Przekazywanie dowolnej liczby argumentów	197
Argumenty pozycyjne i przekazywanie dowolnej liczby argumentów	199
Używanie dowolnej liczby argumentów w postaci słów kluczowych	200
Przechowywanie funkcji w modułach	202
Import całego modułu	202
Import określonych funkcji	203
Użycie słowa kluczowego as w celu zdefiniowania aliasu funkcji	204
Użycie słowa kluczowego as w celu zdefiniowania aliasu modułu	205
Import wszystkich funkcji modułu	205
Nadawanie stylu funkcjom	206
Podsumowanie	207

9

KLASY	209
Utworzenie i użycie klasy	210
Utworzenie klasy Dog	210
Metoda __init__	211
Utworzenie egzemplarza na podstawie klasy	212

Praca z klasami i egzemplarzami	215
Klasa Car	215
Przypisanie atrybutowi wartości domyślnej	216
Modyfikacja wartości atrybutu	217
Dziedziczenie	221
Metoda <code>__init__()</code> w klasie potomnej	221
Definiowanie atrybutów i metod dla klasy potomnej	223
Nadpisywanie metod klasy nadrzędnej	225
Egzemplarz jako atrybut	225
Modelowanie rzeczywistych obiektów	228
Import klas	229
Import pojedynczej klasy	229
Przechowywanie wielu klas w module	231
Import wielu klas z modułu	232
Import całego modułu	233
Import wszystkich klas z modułu	234
Import modułu w module	234
Używanie aliasów	235
Określenie swojego sposobu pracy	236
Biblioteka standardowa Pythona	237
Nadawanie stylu klasom	238
Podsumowanie	239
10	
PLIKI I WYJĄTKI	240
Odczytywanie danych z pliku	241
Wczytywanie całego pliku	241
Względna i bezwzględna ścieżka dostępu do pliku	243
Odczytywanie wiersz po wierszu	244
Praca z zawartością pliku	245
Ogromne pliki, czyli na przykład milion cyfr	246
Czy data Twoich urodzin znajduje się w liczbie pi?	247
Zapisywanie danych w pliku	249
Zapisywanie pojedynczego wiersza	249
Zapisywanie wielu wierszy	249
Wyjątki	251
Obsługiwanie wyjątku <code>ZeroDivisionError</code>	251
Używanie bloku <code>try-except</code>	252
Używanie wyjątków w celu uniknięcia awarii programu	252
Blok <code>else</code>	253
Obsługa wyjątku <code>FileNotFoundError</code>	255
Analiza tekstu	256
Praca z wieloma plikami	257
Ciche niepowodzenie	259
Które błędy należy zgłaszać?	260

Przechowywanie danych	261
Używanie json.dumps() i json.loads()	262
Zapisywanie i odczytywanie danych wygenerowanych przez użytkownika	263
Refaktoryzacja	266
Podsumowanie	269

11

TESTOWANIE KODU	270
Instalowanie pytest za pomocą pip	271
Uaktualnianie pip	271
Instalowanie pytest	272
Testowanie funkcji	272
Test jednostkowy i zestaw testów	274
Zaliczenie testu	274
Wykonywanie testu	275
Niezaliczenie testu	276
Reakcja na niezaliczony test	277
Dodanie nowego testu	279
Testowanie klasy	280
Różne rodzaje metod asercji	280
Klasa do przetestowania	281
Testowanie klasy AnonymousSurvey	283
Używanie danych testowych	285
Podsumowanie	287

CZĘŚĆ II. PROJEKTY 289

12

STATEK, KTÓRY STRZELA POCISKAMI	291
Planowanie projektu	292
Instalacja Pygame	292
Rozpoczęcie pracy nad projektem gry	293
Utworzenie okna Pygame i reagowanie na działania użytkownika	293
Określenie liczby klatek na sekundę	295
Zdefiniowanie koloru tła	296
Utworzenie klasy ustawień	297
Dodanie obrazu statku kosmicznego	298
Utworzenie klasy statku kosmicznego	299
Wyświetlenie statku kosmicznego na ekranie	301
Refaktoryzacja, czyli metody _check_events() i _update_screen()	302
Metoda _check_events()	303
Metoda _update_screen()	303

Kierowanie statkiem kosmicznym	304
Reakcja na naciśnięcie klawisza	305
Umożliwienie nieustannego ruchu	305
Poruszanie statkiem w obu kierunkach	307
Dostosowanie szybkości statku	309
Ograniczenie zasięgu poruszania się statku	310
Refaktoryzacja metody <code>_check_events()</code>	311
Naciśnięcie klawisza Q w celu zakończenia gry	312
Uruchamianie gry w trybie pełnoekranowym	312
Krótkie powtórzenie	313
alien_invasion.py	313
settings.py	314
ship.py	314
Wystrzeliwanie pocisków	315
Dodawanie ustawień dotyczących pocisków	315
Utworzenie klasy <code>Bullet</code>	315
Przechowywanie pocisków w grupie	317
Wystrzeliwanie pocisków	318
Usuwanie niewidocznych pocisków	320
Ograniczenie liczby pocisków	321
Utworzenie metody <code>_update_bullets()</code>	321
Podsumowanie	323
13	
OBCY!	324
Przegląd projektu	325
Utworzenie pierwszego obcego	325
Utworzenie klasy <code>Alien</code>	326
Utworzenie egzemplarza obcego	327
Utworzenie floty obcych	328
Utworzenie rzędów obcych	329
Refaktoryzacja metody <code>_create_fleet()</code>	330
Dodawanie rzędów	332
Poruszanie flotą obcych	334
Przesunięcie obcych w prawo	334
Zdefiniowanie ustawień dla kierunku poruszania się floty	336
Sprawdzenie, czy obcy dotarł do krawędzi ekranu	336
Przesunięcie floty w dół i zmiana kierunku	337
Zestrzeliwanie obcych	339
Wykrywanie kolizji z pociskiem	339
Utworzenie większych pocisków w celach testowych	340
Ponowne utworzenie floty	340
Zwiększenie szybkości pocisku	342
Refaktoryzacja metody <code>_update_bullets()</code>	342

Zakończenie gry	343
Wykrywanie kolizji między obcym i statkiem	343
Reakcja na kolizję między obcym i statkiem	344
Obcy, który dociera do dolnej krawędzi ekranu	347
Koniec gry!	348
Ustalenie, które komponenty gry powinny być uruchomione	349
Podsumowanie	350
14	
PUNKTACJA	351
Dodanie przycisku Gra	351
Utworzenie klasy Button	352
Wyświetlenie przycisku na ekranie	354
Uruchomienie gry	355
Zerowanie gry	356
Dezaktywacja przycisku Gra	357
Ukrycie kursora myszy	357
Zmiana poziomu trudności	358
Zmiana ustawień dotyczących szybkości	359
Wyzerowanie szybkości	360
Punktacja	361
Wyświetlanie punktacji	362
Utworzenie tablicy wyników	363
Uaktualnienie punktacji po zestrzeleniu obcego	364
Zerowanie wyniku	366
Zagwarantowanie uwzględnienia wszystkich trafień	366
Zwiększenie liczby zdobywanych punktów	367
Zaokrąglanie punktacji	368
Najlepsze wyniki	369
Wyświetlenie aktualnego poziomu gry	371
Wyświetlenie liczby statków	375
Podsumowanie	378
15	
GENEROWANIE DANYCH	379
Instalacja matplotlib	380
Wygenerowanie prostego wykresu liniowego	380
Zmienianie etykiety i grubości wykresu	381
Poprawianie wykresu	383
Używanie wbudowanych stylów	384
Używanie funkcji scatter() do wyświetlania poszczególnych punktów i nadawania im stylu	386
Wyświetlanie serii punktów za pomocą funkcji scatter()	387
Automatyczne obliczanie danych	388
Dostosowanie znaczników osi do własnych potrzeb	389

Definiowanie własnych kolorów	390
Użycie mapy kolorów	390
Automatyczny zapis wykresu	392
Błądzenie losowe	392
Utworzenie klasy RandomWalk	392
Wybór kierunku	393
Wyświetlenie wykresu błądzenia losowego	394
Wygenerowanie wielu błędzeń losowych	395
Nadawanie stylu danym wygenerowanym przez błądzenie losowe	396
Symulacja rzutu kością do gry za pomocą plotly	402
Instalacja plotly	402
Utworzenie klasy Die	403
Rzut kością do gry	403
Analiza wyników	404
Utworzenie histogramu	405
Dostosowanie wykresu do własnych potrzeb	406
Rzut dwiema kośćmi	407
Dalsze dostosowanie wykresu do własnych potrzeb	409
Rzut kośćmi o różnej liczbie ścianek	410
Zapisywanie wykresu	411
Podsumowanie	412
16	
POBIERANIE DANYCH	413
Format CSV	414
Przetwarzanie nagłówków pliku CSV	414
Wyświetlanie nagłówków i ich połączenia	415
Wyodrębnienie i odczytanie danych	416
Wyświetlenie danych na wykresie temperatury	417
Moduł datetime	418
Wyświetlanie daty	419
Wyświetlenie dłuższego przedziału czasu	420
Wyświetlenie drugiej serii danych	421
Nakładanie cienia na wykresie	423
Sprawdzenie pod kątem błędów	424
Samodzielne pobieranie danych	428
Mapowanie globalnych zbiorów danych – format GeoJSON	429
Pobranie danych dotyczących trzęsień ziemi	430
Analiza danych GeoJSON	430
Utworzenie listy trzęsień ziemi	433
Wyodrębnienie siły trzęsienia ziemi	433
Wyodrębnienie danych o miejscu wystąpienia trzęsienia ziemi	434
Budowanie mapy świata	435
Przedstawienie siły trzęsienia ziemi	436

Dostosowanie koloru punktu	437
Inne skale kolorów	439
Dodanie tekstu wyświetlanego po wskazaniu punktu na mapie	439
Podsumowanie	441
17	
PRACA Z API	442
Użycie API	442
Git i GitHub	443
Żądanie danych za pomocą wywołania API	443
Instalacja requests	444
Przetworzenie odpowiedzi API	445
Praca ze słownikami odpowiedzi	446
Podsumowanie repozytoriów najczęściej oznaczanych gwiazdką	449
Monitorowanie ograniczeń liczby wywołań API	450
Wizualizacja repozytoriów za pomocą pakietu plotly	451
Nadawanie stylu wykresowi	452
Dodanie własnych podpowiedzi	454
Dodawanie łącz do wykresu	455
Dostosowanie kolorów znaczników do własnych potrzeb	456
Więcej o plotly i API GitHub	457
Hacker News API	457
Podsumowanie	462
18	
ROZPOCZĘCIE PRACY Z DJANGO	463
Przygotowanie projektu	464
Opracowanie specyfikacji	464
Utworzenie środowiska wirtualnego	464
Aktywacja środowiska wirtualnego	465
Instalacja frameworka Django	465
Utworzenie projektu w Django	466
Utworzenie bazy danych	467
Przegląd projektu	468
Uruchomienie aplikacji	469
Definiowanie modeli	470
Aktywacja modeli	471
Witryna administracyjna Django	473
Zdefiniowanie modelu Entry	476
Migracja modelu Entry	477
Rejestracja modelu Entry w witrynie administracyjnej	478
Powtórka Django	479
Tworzenie stron internetowych — strona główna aplikacji	481
Mapowanie adresu URL	482
Utworzenie widoku	484
Utworzenie szablonu	484

Utworzenie dodatkowych stron	486
Dziedziczenie szablonu	486
Strona tematów	489
Strony poszczególnych tematów	492
Podsumowanie	496

19

KONTA UŻYTKOWNIKÓW 497

Umożliwienie użytkownikom wprowadzania danych	497
Dodawanie nowego tematu	498
Dodawanie nowych wpisów	503
Edycja wpisu	507
Konfiguracja kont użytkowników	511
Aplikacja accounts	511
Strona logowania	512
Wylogowanie	515
Strona rejestracji użytkownika	516
Umożliwienie użytkownikom bycia właścicielami swoich danych	520
Ograniczenie dostępu za pomocą dekoratora @login_required	520
Powiązanie danych z określonymi użytkownikami	522
Przyznanie dostępu jedynie odpowiednim użytkownikom	525
Ochrona tematów użytkownika	526
Ochrona strony edit_entry	527
Powiązanie nowego tematu z bieżącym użytkownikiem	528
Podsumowanie	529

20

NADANIE STYLU I WDRÓŻENIE APLIKACJI 531

Nadanie stylu aplikacji Learning Log	532
Aplikacja django-bootstrap5	532
Użycie Bootstrapa do nadania stylu aplikacji Learning Log	533
Modyfikacja pliku base.html	533
Użycie elementu Jumbotron do nadania stylu stronie głównej	540
Nadanie stylu stronie logowania	541
Nadanie stylu stronie tematów	542
Nadanie stylów wpisom na stronie tematu	543
Wdrożenie aplikacji Learning Log	545
Utworzenie konta w Platform.sh	546
Instalacja Platform.sh CLI	546
Instalacja platformshconfig	546
Utworzenie pliku requirements.txt	547
Dodatkowe wymagania dotyczące wdrożenia	547
Dodawanie plików konfiguracyjnych	548
Modyfikacja pliku settings.py dla Platform.sh	552
Użycie Gita do monitorowania plików projektu	553

Utworzenie projektu w Platform.sh	555
Przekazanie projektu do Platform.sh	557
Wyświetlenie wdrożonego projektu	558
Dopracowanie wdrożenia projektu w Platform.sh	558
Utworzenie własnych stron błędów	561
Nieustanna rozbudowa	563
Usunięcie projektu z Platform.sh	564
Podsumowanie	565
A	
INSTALACJA PYTHONA I ROZWIĄZYWANIE PROBLEMÓW	567
Python w Windows	567
Użycie polecenia py zamiast python	568
Ponowna instalacja Pythona	568
Python w systemie macOS	568
Przypadkowa instalacja wersji dostarczonej przez Apple	569
Python 2 w starszych wydaniach systemu macOS	569
Python w systemie Linux	569
Używanie domyślnej instalacji Pythona	569
Instalacja najnowszej wersji Pythona	570
Sprawdzenie aktualnie używanej wersji Pythona	570
Słowa kluczowe Pythona i wbudowane funkcje	571
Słowa kluczowe Pythona	571
Wbudowane funkcje Pythona	571
B	
EDYTORY TEKSTU I ŚRODOWISKA IDE	573
Efektywna praca z VS Code	574
Konfigurowanie VS Code	575
Wybrane skróty klawiszowe VS Code	577
Inne edytory tekstu i środowiska IDE	579
IDLE	579
Geany	579
Sublime Text	579
Emacs i vim	579
PyCharm	580
Notatniki Jupyter Notebooks	580
C	
UZYSKIWANIE POMOCY	581
Pierwsze kroki	581
Spróbuj jeszcze raz	582
Chwila odpoczynku	582
Korzystaj z zasobów tej książki	583

Wyszukiwanie informacji w internecie	583
Stack Overflow	583
Oficjalna dokumentacja Pythona	584
Oficjalna dokumentacja biblioteki	584
r/learnpython	584
Posty na blogach	585
Discord	585
Slack	585

D

UŻYWANIE GITA DO KONTROLI WERSJI	586
Instalacja Gita	587
Konfiguracja Gita	587
Tworzenie projektu	588
Ignorowanie plików	588
Inicjalizacja repozytorium	589
Sprawdzanie stanu	589
Dodawanie plików do repozytorium	590
Zatwierdzanie plików	590
Sprawdzanie dziennika projektu	591
Dругie zatwierdzenie	591
Przywracanie stanu projektu	593
Przywracanie projektu do poprzedniego stanu	594
Usuwanie repozytorium	596

E

ROZWIĄZYWANIE PROBLEMÓW Z WDROŻENIAMI	598
Jak wygląda proces wdrażania?	598
Podstawy rozwiązywania problemów	599
Kierowanie się podpowiedziami wyświetlanymi na ekranie	600
Odczytywanie danych wyjściowych dzienników zdarzeń	601
Rozwiązywanie problemów dotyczących konkretnego systemu operacyjnego	603
Wdrażanie z poziomu systemu Windows	604
Wdrażanie z poziomu systemu macOS	605
Wdrażanie z poziomu systemu Linux	606
Inne podejścia w zakresie wdrożenia	607