

O'REILLY®

WYDANIE V

Projektowanie stron internetowych

PRZEWODNIK DLA POCZĄTKUJĄCYCH
WEBMASTERÓW PO HTML5, CSS3 I GRAFICE

Jennifer Niederst Robbins



Bez względu na to, czy jesteś na początku swojej drogi, czy też jedynie chcesz uporządkować i zaktualizować wiedzę, w tej książce znajdziesz solidne podstawy w zakresie nowoczesnego tworzenia witryn internetowych. Każde zagadnienie zostało przedstawione wizualnie w odpowiednim tempie i jest wspomagane szeregiem ćwiczeń, które umożliwiają wypróbowanie nowych umiejętności. Czytając książkę, po prostu przenosisz się na moje wykłady! – Jennifer Niederst Robbins

Tytuł oryginału: Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, 5th Edition

Tłumaczenie: Agnieszka Górczyńska

ISBN: 978-83-283-6203-1

© 2020 Helion SA

Authorized Polish translation of the English edition of *Learning Web Design, 5th Edition*

ISBN 9781491960202 © 2018 O'Reilly Media, Inc

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

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 Helion SA 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 Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Pliki z przykładami omawianymi w książce można znaleźć pod adresem: <ftp://ftp.helion.pl/przyklady/psippv.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/psippv>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

SPIS TREŚCI

SŁOWO WSTĘPNE	xi
PRZEDMOWA	xiii

CZĘŚĆ I PODSTAWY

1. Rozpoczęcie projektowania witryn internetowych	3
Od czego zacząć?	4
Role podczas tworzenia witryn internetowych	4
Co muszę kupić?	14
Czego się nauczyłeś?	20
Sprawdź się!	20
2. Jak działa internet?	21
Internet a Web	21
Dostarczanie informacji	22
Kilka słów o przeglądarkach	23
Adres strony internetowej (URL)	24
Anatomia strony internetowej	27
Składanie wszystkiego w całość	32
Sprawdź się!	34
3. Kilka ważnych spraw, o których musisz wiedzieć	35
Od przybytku głowa boli	36
Trzymanie się standardów	37
Stopniowe ulepszanie	38

Technika Responsive Web Design	40
Web dla wszystkich, czyli dostępność	41
Need for Speed — liczy się szybkość	44
Sprawdź się!	46

CZĘŚĆ II ZNACZNIKI HTML I STRUKTURA DOKUMENTU

4. Tworzenie prostej strony (przegląd języka HTML)	49
Strona internetowa krok po kroku	49
Przed rozpoczęciem należy uruchomić edytor tekstu	50
Krok 1. Zaczynamy od zawartości	53
Krok 2. Nadajemy dokumentowi strukturę	55
Krok 3. Oznaczamy elementy tekstowe	59
Krok 4. Wstawiamy obrazek	62
Krok 5. Zmieniamy wygląd za pomocą arkusza stylów	66
Kiedy dobre strony nie działają dobrze	67
Walidacja dokumentów	68
Sprawdź się!	70
Przegląd elementów HTML tworzących strukturę dokumentu	70
5. Znaczniki tekstowe	71
Akapity	71
Nagłówki	72
Rozdzielanie treści (pozioma linia)	74
Listy	74
Inne elementy stosowane do treści	78
Struktura zawartości strony	82
Przegląd elementów liniowych	88
Ogólne elementy div oraz span	98
Poprawienie dostępności za pomocą technologii Aria	102
Znaki ucieczki	105
Zestawienie wszystkiego razem	108
Sprawdź się!	111
Przegląd elementów tekstowych	112

6. **Hiperłącza** 113

 Atrybut href 114

 Tworzenie odsyłaczy do stron internetowych 115

 Tworzenie odsyłaczy do stron z własnej witryny 116

 Otwieranie stron docelowych w nowym oknie przeglądarki 126

 Odsyłacze „pocztowe” 127

 Odsyłacze „telefoniczne” 128

 Sprawdź się! 128

 Przegląd elementów — odsyłacze 130

7. **Grafika** 131

 Kilka słów o formatach obrazków 132

 Element img 134

 Dodawanie obrazków SVG 139

 Kod znaczników responsywnych obrazków 146

 Uff! To już koniec 159

 Sprawdź się! 159

 Przegląd elementów — obrazki 162

8. **Tabele** 163

 Jak używać tabel 163

 Podstawowa struktura tabeli 165

 Nagłówki tabel 167

 Tworzenie zakresów komórek 168

 Dostępność tabel 169

 Grupy wierszy i kolumn 171

 Podsumowanie zagadnień związanych z tabelami 173

 Sprawdź się! 175

 Przegląd elementów HTML — tabele 176

9. **Formularze** 177

 Jak działają formularze 177

 Element form 179

 Zmienne oraz zawartość 182

 Wielkie podsumowanie kontrolek 183

 Dostępność formularzy 203

Projekt i układ formularza	206
Sprawdź się!	208
Formularze — przegląd elementów	209
10. Osadzone treści multimedialne	215
Okno w oknie (element iframe)	215
Osadzanie różnych treści na stronie	218
Elementy video i audio	219
Element canvas	228
Sprawdź się!	233
Osadzone treści multimedialne — przegląd elementów	234

CZĘŚĆ III CSS I PREZENTACJA DOKUMENTU

11. Kaskadowe arkusze stylów	239
Zalety CSS-a	239
Jak działają arkusze stylów	240
Najważniejsze koncepcje	246
Jednostki miar w CSS-ie	253
Narzędzia programistyczne wbudowane w przeglądarkę WWW	256
Dalsza nauka CSS-a	258
Sprawdź się!	259
12. Formatowanie tekstu (i jeszcze więcej selektorów)	261
Podstawowe właściwości czcionek	262
Typografia zaawansowana wykorzystująca CSS3	277
Zmiana koloru tekstu	280
Więcej typów selektorów	281
Zmiana stylu wiersza tekstu	287
Podkreślenia oraz inne „dekoracje”	290
Zmiana wielkości liter	291
Odstępy	292
Cienie pod tekstem	293
Zmiany wypunktowania i numeracji list	296
Sprawdź się!	299
Przegląd CSS-a — właściwości dotyczące czcionki oraz tekstu	301

13. Kolory i tła	303
Określanie wartości koloru	303
Kolor pierwszego planu	311
Kolor tła	312
Wycinanie tła	314
Zabawy z przezroczystością	315
Wprowadzenie do selektorów pseudoklas	316
Selektory pseudoelementów	320
Selektory atrybutów	323
Obrazki tła	324
Skrótowa właściwość background	338
Prawie tęcza (gradienty)	340
I wreszcie — zewnętrzne arkusze stylów	348
Zakończenie rozdziału	351
Sprawdź się!	352
Przegląd CSS-a — właściwości dotyczące koloru oraz tła	354
14. Model pojemnika	355
Pojemnik elementu	355
Określanie wymiarów zawartości elementu	356
Dopełnienie	361
Obramowanie	366
Marginesy	376
Przypisywanie ról wyświetlania	380
Dodawanie cienia do elementów	382
Sprawdź się!	384
Przegląd CSS-a — podstawowe właściwości modelu pojemnika	384
15. Pływanie oraz pozycjonowanie	387
Normalny układ dokumentu	387
Pływanie	388
Efektowne opływanie tekstu z użyciem kształtów CSS	399
Podstawy pozycjonowania	405
Pozycjonowanie względne	407
Pozycjonowanie bezwzględne	408
Pozycjonowanie sztywne	416

Sprawdź się!	417
Przegląd CSS-a — właściwości dotyczące pływania i pozycjonowania	418
16. Układ CSS z użyciem modeli Flexbox i siatki	419
Elastyczne pojemniki w modelu CSS Flexbox	419
Układ oparty na siatce CSS	447
Sprawdź się	479
Przegląd CSS-a — właściwości układu	482
17. Responsywny układ strony	485
Dlaczego RWD?	485
Przepis na układ responsywny	486
Wybieranie punktów granicznych	495
Tworzenie projektu responsywnego	499
Kilka słów o testowaniu	512
Więcej zasobów RWD	514
Sprawdź się!	516
18. Przekształcenia, transformacje i animacje	517
Przejścia CSS	517
Transformacje CSS	527
Animacja klatkowa	536
Zakończenie	542
Sprawdź się!	542
Przegląd właściwości CSS-a — przejścia, transformacje i animacje	545
19. Więcej technik CSS	547
Nadawanie stylów formularzom	547
Nadawanie stylów tabelom	550
Czysta strona (zerowanie stylów CSS i normalize.css)	554
Techniki zastępowania tekstu obrazkiem	556
Technika CSS Sprites	557
Wykrywanie obsługiwanych funkcji CSS	559
Podsumowanie arkuszy stylów	564
Sprawdź się!	564
Przegląd CSS-a — właściwości dotyczące tabel oraz list	566

20. Nowoczesne narzędzia do tworzenia witryn internetowych 567

- Krótkie wprowadzenie do powłoki 567
- Potężne narzędzia CSS-a (procesory) 572
- Narzędzia do kompilacji (Grunť i Gulp) 578
- System kontroli wersji za pomocą Git i GitHub 581
- Podsumowanie 588
- Sprawdź się! 589

CZĘŚĆ IV JAVASCRIPT

21. Wprowadzenie do JavaScriptu 593

- Czym jest JavaScript? 593
- Dodawanie skryptów na stronę 597
- Anatomia skryptu 598
- Obiekt przeglądarki 613
- Zdarzenia 613
- Podsumowanie 616
- Więcej informacji o JavaScriptcie 617
- Sprawdź się! 619

22. Korzystanie z JavaScriptu i modelu obiektowego DOM 621

- Poznaj model DOM 621
- Wypełniacze 630
- Javascriptowe biblioteki 632
- Wielkie zakończenie 637
- Sprawdź się! 637

CZĘŚĆ V TWORZENIE GRAFIKI STRON INTERNETOWYCH

23. Podstawy grafiki stron internetowych 641

- Źródła obrazków 641
- Poznaj formaty grafiki 644
- Rozmiar oraz rozdzielczość grafiki 658
- Strategia związana z obrazkami 662
- Ikona favicon 667

Podsumowanie informacji dotyczących grafiki stron internetowych	670
Sprawdź się!	670
24. Tworzenie obrazków na strony internetowe	671
Zapisywanie obrazka w formacie internetowym	671
Praca z przezroczystością	676
Wskazówki dotyczące tworzenia obrazków responsywnych	680
Optymalizacja obrazków	691
Sprawdź się!	701
25. SVG	703
Rysowanie za pomocą języka XML	705
Funkcjonalność SVG jako XML	713
Narzędzia SVG	718
Wskazówki dotyczące tworzenia plików SVG	720
Responsywne obrazki SVG	724
Dalsza lektura	731
Sprawdź się!	732
Gotowe!	733
 CZĘŚĆ VI DODATKI	
A. Odpowiedzi do ćwiczeń	737
B. Atrybuty globalne HTML5	753
C. Selektory CSS Level 3 i Level 4	755
D. Od HTML+ do HTML5	759

JAK DZIAŁA INTERNET?

Projektowaniem stron internetowych zajęłam się na początku 1993 roku — w czasie dość nieodległym od początków samego Internetu. Od tamtej chwili minęło już ćwierć wieku, a mimo to wciąż doskonale pamiętam, kiedy pierwszy raz zobaczyłam stronę internetową. Nie rozumiałam wtedy, skąd pochodzą te informacje i w jaki sposób to wszystko działa.

W tym rozdziale poznasz podstawową terminologię, z jaką możesz się spotkać, kiedy mowa o internecie. Zaczniemy od ogólnych informacji, a później skupimy się na szczegółach.

INTERNET A WEB

Nie, to nie jest spór o pryncypia, a jedynie okazja do wskazania istniejącego rozróżnienia pomiędzy tymi dwoma pojęciami, które coraz częściej używane są wymiennie.

Internet to sieć połączonych komputerów. Nie jest on własnością żadnej firmy — jest przedsięwzięciem zarządzanym przez standardy oraz reguły. Celem połączenia komputerów ze sobą jest oczywiście wymiana informacji. Istnieje wiele sposobów przekazywania informacji między komputerami, takich jak poczta elektroniczna (POP3/IMAP/SMTP), transfer plików (FTP), bezpieczna powłoka (SSH), ale też wiele wyspecjalizowanych metod leżących u podstaw internetu. Te ustandaryzowane metody przekazywania danych lub dokumentów za pośrednictwem sieci nazywane są **protokołami**.

Web (czyli tak naprawdę World Wide Web — stąd skrót „www” w adresach stron) to tylko jeden ze sposobów wymiany informacji w internecie. Jego cechą charakterystyczną jest to, że pozwala na łączenie dokumentów za pomocą odsyłaczy **hipertekstowych** (ang. *hypertext links*), dzięki czemu stało się możliwe tworzenie sieci połączonych ze sobą informacji. WWW wykorzystuje protokół **HTTP** (ang. *HyperText Transfer Protocol*). Akronim ten wygląda znajomo — widzisz go na początku większości adresów stron internetowych (zajmiemy się tym dokładniej w kolejnym podrozdziale).

W TYM ROZDZIALE:

Wyjaśnienie tego, czym jest Web i jak się ma do internetu

Rola serwera

Rola przeglądarki

Wprowadzenie do adresów URL oraz ich składników

Anatomia strony internetowej

Web jest elementem internetu. To tylko jeden ze sposobów przekazywania informacji między komputerami połączonymi w sieć.

Krótką historia World Wide Web

WWW narodziło się w laboratorium fizyki cząstek CERN w Genewie (Szwajcaria) w 1989 roku. Tam specjalista od komputerów Tim Berners-Lee jako pierwszy zaproponował sieciowy system zarządzania informacją, wykorzystujący hipertekst do łączenia powiązanych ze sobą dokumentów. Wraz ze swoim kolegą Robertem Cailliau stworzył prototyp i opublikował go, by poznać opinie innych. Przez pierwszych kilka lat strony internetowe zawierały tylko tekst. Trudno uwierzyć, że w 1992 roku na całym świecie działało tylko pięćdziesiąt serwerów WWW.

Prawdziwe przyspieszenie wzrostu popularności WWW nastąpiło po 1992 roku, głównie za sprawą pierwszej graficznej przeglądarki (NCSA Mosaic). Dzięki temu stało się możliwe przejście WWW ze świata nauki do świata mediów. Rozwój technologii webowych jest nadzorowany przez World Wide Web Consortium (W3C).

Jeśli chcesz dokładniej poznać historię sieci, zajrzyj na stronę *A Little History of the World Wide Web* (www.w3.org/History.html).

CIEKAWOSTKA: na wymienionej stronie można zobaczyć, że w lipcu 1993 roku odbyły się pierwsze warsztaty dla projektantów witryn internetowych. Wprawdzie nie wzięłam w nich udziału, ale *zaprojektowałam* pamiątkową koszulkę.

DOSTARCZANIE INFORMACJI

Poświęćmy teraz chwilę komputerom, dzięki którym istnieje internet. Ponieważ „serwują” one dokumenty na żądanie, zostały nazwane **serwerami**. Serwer to tak naprawdę oprogramowanie (a nie sam komputer), które pozwala komputerowi na komunikację z innymi komputerami. Przyjęło się jednak, że słowa „serwer” używa się również w odniesieniu do samego komputera. Rola oprogramowania serwera polega na oczekiwaniu na żądanie informacji, pobraniu tej informacji i jak najszybszym jej odesłaniu.

W samych komputerach nie ma nic specjalnego — i to nieważne, czy myślisz o maszynie o dużej mocy obliczeniowej opartej na Uniksie, czy o zwykłym komputerze osobistym. Za wszystko odpowiedzialne jest oprogramowanie serwera. By komputer mógł być częścią internetu, musi na nim działać oprogramowanie serwera WWW umożliwiające obsługę protokołu HTTP. Serwery WWW są również nazywane **serwerami HTTP**.

Dostępnych jest wiele serwerów WWW, z czego dwoma najpopularniejszymi są Apache (oprogramowanie typu **open source**) oraz Microsoft Internet Information Services (IIS). Serwer Apache jest dostępny dla komputerów z systemami operacyjnymi opartymi na Uniksie oraz na Macach z systemem macOS, ale istnieje również wersja przeznaczona dla systemu Windows. Microsoft IIS jest z kolei częścią rodziny rozwiązań serwerowych tej firmy.

Każdy komputer i urządzenie (takie jak modem, router czy smartfon) podłączone do internetu ma przypisany unikalny liczbowy **adres IP** (IP to skrót od *Internet Protocol*). Na przykład w chwili gdy piszę te słowa, komputer udostępniający stronę oreilly.com ma adres IP 199.27.145.64. Posługiwanie się adresami w takiej postaci mogłoby przypaść każdemu o ból głowy, dlatego na szczęście powstał system **DNS** (ang. *Domain Name System*), który pozwala na odwołanie się do serwera również za pomocą **nazwy domeny** (oreilly.com). Liczbowy adres IP jest użyteczny dla komputerów, natomiast nazwa domeny jest bardziej czytelna dla ludzi. Dopasowanie tekstowych nazw domen do odpowiadających im adresów IP jest

■ TERMINOLOGIA

Open source

Oprogramowanie open source (nazywane również oprogramowaniem o otwartym kodzie źródłowym) jest tworzone zwykle przez większą grupę programistów, a jego kod źródłowy jest dostępny dla innych programistów, którzy mogą go używać i modyfikować. Programy open source są zwykle dostępne za darmo.

zadaniem osobnego **serwera DNS**. Jeżeli adres IP porównasz do numeru telefonu, serwer DNS mógłbyś porównać do książki telefonicznej.

Możliwe jest skonfigurowanie serwera WWW w taki sposób, by więcej niż jedna domena była odwzorowywana na jeden adres IP, co pozwala wielu stronom na współdzielenie jednego serwera.

KILKA SŁÓW O PRZEGLĄDARKACH

Wiemy już, że zadaniem serwera jest dostarczanie informacji na żądanie. A gdzie druga strona? Oprogramowanie wykonujące żądanie jest nazywane **klientem**. W przypadku dokumentów WWW rolę klienta spełniają przeglądarki (desktopowe i mobilne), a także dodatkowe oprogramowanie (takie jak na przykład czytniki stron). Przeglądarka (która w tym kontekście w kręgach technicznych jest nazywana **user-agent**) zajmuje się wyświetleniem tego, co otrzymała z serwera.

Żądania oraz odpowiedzi są transportowane za pomocą wspomnianego wcześniej protokołu HTTP. Choć mówimy o dokumentach, protokół HTTP może być wykorzystywany do transferu plików obrazów, filmów i dźwięków, a także innych zasobów składających się na stronę lub aplikację internetową.

Kiedy mowa o przeglądarce, zazwyczaj mamy przed oczami okno na monitorze komputera, w którym jest wyświetlana strona internetowa. Takie przeglądarki są zwykle nazywane przeglądarkami desktopowymi lub graficznymi. W czasie gdy ta książka powstawała, najpopularniejsze z nich to Chrome, Edge, Firefox, Internet Explorer, Opera, Safari i Vivaldi.

Jednak obecnie ponad połowa ruchu sieciowego pochodzi z przeglądarek WWW działających w urządzeniach mobilnych: smartfonach, tabletach itd. Do tych przeglądarek zaliczamy Safari dla systemu iOS, Android i Chrome w systemie Android, Opera Mini i wiele innych zainstalowanych domyślnie w pozostałych urządzeniach mobilnych. (Pełną listę przeglądarek WWW dla urządzeń mobilnych znajdziesz na stronie https://en.wikipedia.org/wiki/Mobile_browser). Przeglądanie zasobów internetu za pomocą ekranu dotykowego stało się obecnie normą.

Podczas projektowania i tworzenia stron należy uwzględniać wiele różnych możliwości ich odbioru. Niedowidzący użytkownicy mogą na przykład odsłuchiwać strony internetowe za pomocą czytnika ekranu (lub powiększać stronę tak, by tekst był jak największy). Użytkownicy mający problemy z poruszaniem się mogą używać dodatkowych urządzeń pomagających im wpisywać tekst i nawigować po stronie. Tworzone strony muszą być dostępne i użyteczne dla wszystkich użytkowników, niezależnie od sposobu, w jaki z nich korzystają.

Należy pamiętać, że strony internetowe mogą wyglądać i działać w różny sposób nawet w najnowszych przeglądarkach. Jest to spowodowane przede wszystkim różnym stopniem obsługi technologii webowych, ale również możliwościami i konfiguracją komputerów poszczególnych użytkowników.

■ TERMINOLOGIA

Intranet i ekstranet

Kiedy mówimy o witrynie internetowej, zazwyczaj zakładamy, że jest ona dostępna dla każdej osoby używającej internetu. Wiele firm korzysta jednak z zalet tego wspaniałego środka do wymiany informacji tylko w ramach własnej organizacji. Tego typu sieci są nazywane **intranetami**. Witryny intranetowe są zbudowane i działają w ten sam sposób jak witryny internetowe, ale są dostępne jedynie w wewnętrznej sieci firmy (dzięki **zaporom sieciowym** — ang. **firewalls**). Intranety mają wiele zastosowań, na przykład pozwalają na współdzielenie informacji o pracownikach czy danych księgowych.

Ekstranet jest podobny do intranetu, jednak umożliwia dostęp wybranym użytkownikom spoza firmy. Na przykład firma produkująca jakiś towar może udostępnić swoim klientom hasła, dzięki którym będą mogli sprawdzać status swoich zamówień w bazie danych firmy. Oczywiście dane, do których klient ma dostęp, są ograniczone.

■ TERMINOLOGIA

Server-side i client-side

W świecie aplikacji webowych często można się zetknąć z terminami „server-side” i „client-side”. Pojęcia te wskazują, po której „stronie” dochodzi do przetwarzania danych. Aplikacje client-side działają po stronie klienta, czyli na komputerze użytkownika (tzw. **frontend**), natomiast aplikacje oraz funkcje server-side wykorzystują moce przetwarzania komputera serwera (tzw. **backend**).

Silniki generujące treść w przeglądarkach WWW

Program odpowiedzialny za konwersję kodu HTML i CSS na treść wyświetlaną na ekranie to tzw. [silnik przeglądarki WWW](#). Przeglądarki używane w tradycyjnych komputerach i urządzeniach mobilnych składają się z silników generujących treść oraz kodu dodatkowego, zapewniającego interfejs użytkownika i pewną funkcjonalność. Wprawdzie w książce napisałam wiele o tym, które przeglądarki WWW obsługują określone funkcje, ale technicznie rzecz biorąc, odwoływałam się wówczas do silników tych przeglądarek. Znaczna część przeglądarek WWW korzysta z jednego

silnika. Przykładowo silnik Blink znalazł zastosowanie w przeglądarkach Chrome, Opera i różnych przeglądarkach w systemie Android. W tabeli 2.1 wymieniłam silniki używane przez obecnie najpopularniejsze przeglądarki WWW. Więcej informacji na ten temat znajdziesz na przykład w Wikipedii na stronach https://en.wikipedia.org/wiki/Comparison_of_browser_engines i https://en.wikipedia.org/wiki/Comparison_of_web_browsers.

TABELA 2.1. Najpopularniejsze obecnie przeglądarki WWW i stosowane w nich silniki

Przeglądarka WWW	Silnik
Chrome 28+	Blink (powstały na podstawie silnika WebKit)
Firefox (wszystkie wersje)	Gecko (z wyjątkiem Firefoksa dla systemu iOS, która to przeglądarka używa silnika WebKit)
Safari i Safari dla iOS (wszystkie wersje)	WebKit
Internet Explorer 4–11	Trident
MS Edge (wszystkie wersje)	EgdyHTML (powstały na podstawie silnika Trident)
Opera 15+	Blink (powstały na podstawie silnika WebKit)

Rzeczywistość jest taka, że wygląd i sposób działania strony internetowej mogą zależeć od przeglądarki WWW. To wynika z różnic w obsłudze poszczególnych technologii internetowych, odmiennych możliwości urządzeń, a także zmian w ustawieniach przeglądarek wprowadzanych przez użytkowników. To jest najtrudniejszy aspekt projektowania i tworzenia treści dla internetu.

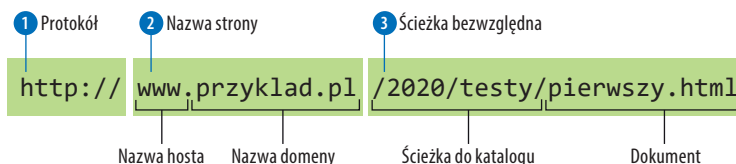
ADRES STRONY INTERNETOWEJ (URL)

Każda strona i każdy zasób mają specjalny adres nazywany URL (ang. *Uniform Resource Locator*). Adresy te są dla nas czymś naturalnym — widzimy je na co dzień na wizytówkach, autobusach i w reklamach telewizyjnych. Obecnie adresy stron internetowych na stałe zagościły w języku potocznym.

Niektóre adresy URL są krótkie i ładne. Inne mogą wyglądać jak dziwne łańcuchy znaków rozdzielone kropkami i ukośnikami, jednak każdy element adresu URL ma swoje przeznaczenie. Aby się o tym przekonać, rozbierzmy jeden z nich na części.

Części adresu URL

Pełny adres URL składa się z trzech głównych części: protokołu, nazwy strony oraz ścieżki bezwzględnej do dokumentu lub zasobu, co widać na rysunku 2.1.



RYSUNEK 2.1. Części adresu URL

- 1 `http://`
Pierwszy element adresu URL definiuje protokół wykorzystywany w tej konkretnej transakcji. Jeśli jest to *http*, serwer jest informowany o tym, że ma użyć protokołu HTTP czy inaczej: przejść w „tryb WWW”. Czasami adres URL rozpoczyna się od *https://*, co wyjaśnię w ramce „HTTPS — protokół bezpiecznej sieci” w dalszej części rozdziału.
- 2 `www.przyklad.pl`
Kolejna część adresu URL identyfikuje witrynę internetową po nazwie domeny. W tym przykładzie nazwa domeny to *przyklad.pl*. Część *www* znajdująca się na początku jest określoną nazwą hosta w tej domenie. Taka nazwa hosta (*www*) stała się konwencją, jednak nie jest regułą, więc czasami można ją pominąć. W domenie może się znajdować więcej niż jedna witryna internetowa (mówi się wtedy o subdomenach), na przykład: *programisci.przyklad.pl*, *klienci.przyklad.pl* itd.
- 3 `/2020/testy/pierwszy.html`
To ścieżka bezwzględna żadanego dokumentu HTML, czyli *pierwszy.html*. Słowa rozdzielone ukośnikami określają ścieżkę przechodzącą przez kolejne poziomy katalogów — rozpoczynają się od katalogu głównego hosta aż po sam plik *pierwszy.html*. Ponieważ internet początkowo składał się z komputerów działających pod systemem operacyjnym UNIX, w świecie WWW zachowało się wiele reguł oraz konwencji z Uniksa (na przykład zastosowanie ukośnika /).

Podsumowując, przykładowy adres URL z rysunku 2.1 oznacza, że przeglądarka chce wykonać protokół HTTP w celu połączenia się z serwerem WWW o nazwie *www.przyklad.pl* i zażądania dokumentu *pierwszy.html* (znajdującego się w podkatalogu *testy*, który z kolei mieści się w katalogu *2020*).

Uprozczone adresy URL

Oczywiście nie wszystkie adresy URL, które napotkasz, są tak długie, jak mogłeś wcześniej zobaczyć na przykładzie. Jeżeli chcesz przejść do witryny wydawnictwa Helion, wystarczy że wpiszesz *helion.pl* zamiast *http://www.helion.pl/index.html*. Teraz zobaczysz, jak to działa.

URL kontra URI

Konsorcjum W3C i społeczność programistów odchodzą od terminu URL (ang. *uniform resource locator*) na rzecz bardziej ogólnego i pod względem technicznym odpowiedniejszego URI (ang. *uniform resource identifier*). Jednak zarówno na ulicy, jak i w firmie nadal będziesz słyszał URL.

Oto jeden z argumentów URL kontra URI: URL to po prostu jeden z typów URI identyfikujących zasób na podstawie jego położenia (litera L w skrócie URL) w sieci. Innym typem URI jest URN, który identyfikuje zasób na podstawie nazwy lub przestrzeni nazw (litera N w skrócie URN).

Ponieważ nazwa URL jest bardziej rozpowszechniona, będę ją stosowała w tej książce. Musisz po prostu zapamiętać, że URL jest podzbiorem URI, a te pojęcia są często używane wymiennie.

Jeżeli jednak chcesz jeszcze dźżyć ten temat, zajrzyj do artykułu w Wikipedii na stronie https://pl.wikipedia.org/wiki/Uniform_Resource_Identifier.

HTTPS — protokół bezpiecznej sieci

Jeżeli podczas robienia zakupów w internecie lub korzystania z bankowości internetowej spojrzysz na pasek adresu strony, zauważysz, że używany jest protokół HTTPS. Tutaj litera **S** oznacza **secure**, czyli bezpieczeństwo. HTTPS jest zmodyfikowaną wersją protokołu HTTP, stosującą szyfrowanie informacji podczas ich przekazywania między klientem a serwerem. Każda strona internetowa zawierająca formularz HTML akceptujący tekst, na przykład pole wyszukiwania lub logowania, powinna używać protokołu HTTPS.

W czasie gdy ta książka powstawała, około 60% stron (a ta liczba wciąż się zwiększa) stosowało protokół HTTPS i były ku temu ważne powody. Nie tylko dobrze jest zapewnić bezpieczeństwo danych użytkowników podczas ich przekazywania, ale jeszcze Google na różne sposoby zachęca do stosowania protokołu HTTPS. Przykładowo, jeśli masz witrynę internetową akceptującą tekstowe dane wejściowe i nie używasz HTTPS, Twoja witryna nie będzie zajmowała wysokiego miejsca w wynikach wyszukiwania Google. Ponadto w przeglądarce WWW taka witryna będzie w pasku górnym okna przeglądarki oznaczona jako niebezpieczna.

HTTPS działa w połączeniu z innym protokołem, SSL (ang. **secure socket layer**), który musi być włączony w serwerze, aby można było stosować bezpieczne transakcje. Firmy hostingowe oferują możliwość włączenia SSL, często bezpłatnie.

Warto w tym miejscu dodać, że protokół HTTPS chroni dane podczas ich przekazywania do serwera i nie ma nic wspólnego z „zabezpieczeniem” witryny internetowej przed hakerami.

Pominięcie protokołu

Skoro niemal wszystkie strony internetowe używają protokołu HTTP, człon <http://> jest często uznawany za domyślny. Z taką sytuacją mamy do czynienia w przypadku witryn podawanych w reklamach (magazyny, telewizja itd.), ponieważ krótszy adres jest znacznie łatwiejszy do zapamiętania.

Ponadto przeglądarki WWW zostały zaprogramowane do automatycznego dodawania członu <http://>, co należy uznać za udogodnienie skracające czas wpisywania adresu. Może się wydawać, że protokół został pominięty, ale tak naprawdę został w tle przekazany do serwera.

Gdy w rozdziale 6. zaczniesz używać adresów URL do tworzenia hiperłączy w dokumentach HTML, przekonasz się, że podanie protokołu w łączy jest niezbędne, aby zdefiniować połączenie ze stroną internetową znajdującą się na innym serwerze.

Pliki domyślne

Wiele adresów nie zawiera nazw plików, ale wskazuje na katalog, jak poniższe:

<http://www.helion.pl>

<http://www.przyklad.pl/zestawienie/>

Kiedy serwer otrzymuje żądanie z adresem katalogu, a nie określonego pliku, szuka w tym katalogu domyślnego dokumentu, zazwyczaj noszącego nazwę *index.html*, i odsyła do przeglądarki właśnie ten dokument. Kiedy zatem wpisze się powyższe adresy URL w przeglądarce, tak naprawdę użytkownik przeglądarki zobaczy następujące strony:

<http://www.helion.pl/index.html>

<http://www.przyklad.pl/zestawienie/index.html>

Nazwa pliku domyślnego (czyli **pliku indeksu**) może się różnić i zależy od tego, jak skonfigurowany jest serwer. W powyższych przykładach plik nosi nazwę *index.html*, jednak niektóre serwery używają na przykład pliku o nazwie *default.htm*. Jeśli strony są generowane po stronie serwera, domyślny plik może się nazywać *index.php* lub *default.aspx*. Jeśli nie wiesz, jak nazwać domyślny plik, zapytaj administratora serwera.

W pierwszym z powyższych przykładów można zauważyć, że oryginalny adres URL nie miał końcowego ukośnika, który wskazywałby, że jest to katalog. Kiedy ukośnik zostaje pominięty, serwer dodaje go, jeśli znajdzie katalog o danej nazwie. Jeżeli podany element faktycznie jest katalogiem, serwer prosi przeglądarkę WWW o ponowne wykonanie żądania, tym razem wraz z ukośnikiem na końcu. Ostatecznie ukośnik i tak znajduje się po nazwie katalogu, nawet jeśli na początku nie został podany (zobacz ramkę „Wskazówka dotycząca wydajności działania”).

Domyślny plik jest również przydatny ze względów bezpieczeństwa. Niektóre serwery (w zależności od konfiguracji), jeśli nie znajdą domyślnego pliku, zwracają przeglądarce zawartość katalogu. Na rysunku 2.2 widać okno przeglądarki, w którym są wyświetlone wszystkie pliki znajdujące się w katalogu *prywatne-zdjecia*, a to wszystko ze względu na brak domyślnego pliku. Jednym ze sposobów na zapobiegnięcie nieuprawnionemu przeglądaniu plików znajdujących się na serwerze jest umieszczenie w każdym katalogu pliku indeksu. Administrator systemu może również dodać inne zabezpieczenia, by zawartość naszych katalogów nie była wyświetlana w przeglądarce.

ANATOMIA STRONY INTERNETOWEJ

Wszyscy wiemy, jak wygląda strona internetowa wyświetlona w przeglądarce, jednak co tak naprawdę „siedzi” w środku?

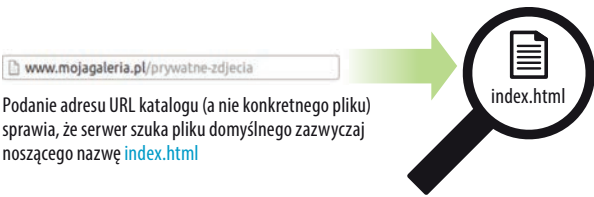
Na górze rysunku 2.3 widać prostą stronę internetową wyświetloną w oknie przeglądarki. Choć można ją zobaczyć jako jeden spójny dokument, tak naprawdę składa się z czterech niezależnych plików: dokumentu HTML (*index.html*), arkusza stylów (*kitchen.css*) oraz dwóch plików graficznych (*foods.gif* oraz *spoon.gif*). Dokument HTML zarządza wszystkim.

Dokumenty HTML

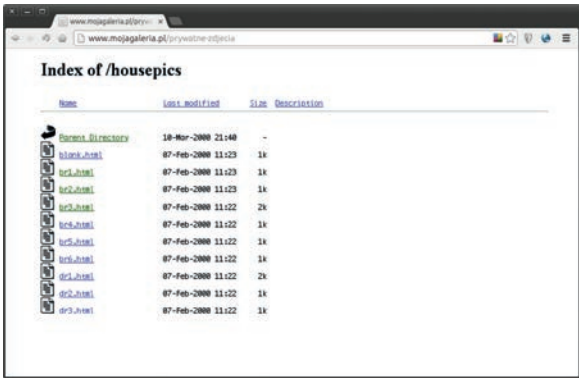
Większość osób będzie pewnie zaskoczona — jak ja sama kiedyś — że bogate w grafikę oraz interaktywne strony internetowe są generowane na podstawie prostych dokumentów tekstowych. Te pliki tekstowe określa się mianem **dokumentów źródłowych**.

Przyjrzyjmy się plikowi *index.html* — dokumentowi tworzącemu stronę internetową Kuchni Kasi. Można zauważyć, że znajduje się w nim zawartość tekstowa strony wraz ze specjalnymi **znacznikami** (oznaczonymi nawiasami ostrymi: `<` oraz `>`) opisującymi każdy element strony.

Dodawanie znaczników opisujących dokument znane jest jako „oznaczenie” dokumentu. Strony internetowe wykorzystują język znaczników znany jako **HyperText Markup Language** (w skrócie **HTML**), który został utworzony specjalnie dla dokumentów zawierających odсылce hipertekstowe. HTML definiuje dziesiątki elementów tekstowych, które składają się na dokument, na przykład nagłówki, akapity, tekst zaakcentowany, a także oczywiście odсылce. Istnieją również elementy HTML dodające informacje o samym dokumencie (na przykład jego tytuł), wstawiające na stronę media, takie jak obrazki czy filmy wideo, a także różnego typu widżety.



Niektóre serwery skonfigurowane są w taki sposób, by wyświetlać zawartość katalogu, jeśli plik domyślny nie zostanie odnaleziony



RYСУNEK 2.2. Niektóre serwery wyświetlają zawartość katalogu, jeśli domyślny plik nie zostanie odnaleziony

■ WSKAZÓWKA DOTYCZĄCA WYDAJNOŚCI DZIAŁANIA

Jeżeli chcesz zmniejszyć liczbę żądań wykonywanych do serwera, w adresach URL łącz zawsze umieszczaj ukośnik na końcu nazwy katalogu.



Strona wyświetlona w oknie przeglądarki składa się tak naprawdę z czterech plików: dokumentu HTML, arkusza stylów oraz dwóch obrazków. Znaczniki w dokumencie HTML są instrukcjami dla przeglądarki, które określają strukturę tekstu oraz lokalizację obrazków

index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Kuchnia Kasi</title>
<link rel="stylesheet" href="kitchen.css" type="text/css" >
</head>

<body>
<h1>Kuchnia Kasi</h1>

<p>Jeśli lubisz czytać o <strong>gotowaniu i jedzeniu</strong>, chciałbyś poznać
niektóre z najlepszych restauracji na świecie lub po prostu szukasz ciekawych
przepisów do swojej kolekcji, <em>z pewnością spodoba Ci się ta strona!</em></p>

<p> Pozdrawiam, Kasia z Kuchni Kasi</p>
<hr>
<p><small>&copy; 2020, Kasia Kuchta</small></p>
</body>
</html>
```

kitchen.css

```
body { font: normal 1em Verdana; margin: 1em 10%;}
h1 { font: italic 3em Georgia; color: rgb(23, 109, 109); margin: 1em 0 1em;}
img { margin: 0 20px 0 0; }
h1 img { margin-bottom: -20px; }
small { color: #666666; }
```

foods.gif



spoon.gif



RYSUNEK 2.3. Plik źródłowy oraz obrazki składające się na prostą stronę internetową

Można wyświetlić kod źródłowy każdej strony; nieco informacji na ten temat znajdziesz w ćwiczeniu 2.1.

Obecnie używana wersja języka HTML to HTML5. Od czasu powstania HTML-a udostępniono już kilka jego wersji, z których część pozostaje w użyciu. Pełną historię języka HTML we wszystkich jego wersjach oraz przedstawienie cech, dzięki którym HTML5 jest tak unikatowy, znajdziesz w dodatku D.

ĆWICZENIE 2.1. Zobacz źródło

Plik HTML każdej strony możesz wyświetlić w postaci jej kodu źródłowego w przeglądarce WWW. Większość nowoczesnych przeglądarek nadal ma funkcję pozwalającą na wyświetlenie źródła strony za pomocą narzędzi programistycznych. Dokument wraz ze źródłem strony jest najczęściej wyświetlany w oddzielnym oknie lub panelu programistycznym w dolnej części bieżącego okna.

Oto informacje, jak znaleźć funkcję wyświetlającą kod źródłowy strony w najważniejszych przeglądarkach WWW:

Safari — opcja menu **Programowanie/Pokaż kod źródłowy strony**.

Chrome — kliknij stronę prawym przyciskiem myszy, a następnie z menu kontekstowego wybierz opcję **Widok/Programista/Wyświetl źródło strony**. Ewentualnie możesz użyć skrótu **Ctrl+U**.

Firefox — opcja menu **Narzędzia/Dla twórców witryn/Źródło strony**.

MS Edge — prawym przyciskiem myszy kliknij stronę, a następnie z menu kontekstowego wybierz opcję **Pokaż źródło**. Jeżeli nie widzisz tej opcji w menu kontekstowym, musisz ją włączyć w tzw. **ustawieniach dewelopera**. Otwórz nowe okno przeglądarki, a następnie w pasku adresu wpisz **about : flags**. W sekcji **Ustawienia dewelopera** zaznacz opcję **Pokaż polecenia „Pokaż źródło” i „Sprawdź element” w menu kontekstowym**. Jeżeli teraz przejdziesz do strony internetowej i klikniesz ją prawym przyciskiem myszy, zobaczysz opcję **Pokaż źródło**. Zawsze możesz również skorzystać ze skrótu **Ctrl+U** lub nacisnąć klawisz **F12**.

1. W wybranej przeglądarce WWW wpisz następujący adres URL:

<https://www.learningwebdesign.com/5e/kitchen.html>

Zobaczysz angielską wersję strony przedstawionej na rysunku 2.3.

2. Zastosuj zamieszczone wcześniej informacje o sposobie wyświetlenia kodu źródłowego strony. Powinien być dokładnie taki sam jak na rysunku 2.3.
3. Jeżeli chcesz wyświetlić nieco bardziej skomplikowaną stronę, przejdź do strony głównej witryny <https://www.learningwebdesign.com/> i wyświetl jej źródło.
4. Kod źródłowy większości stron jest znacznie bardziej skomplikowany. Jeśli chcesz się o tym przekonać, wyświetl na przykład kod źródłowy strony www.helion.pl. Znajdziesz na niej arkusze stylów, skrypty, osadzoną grafikę SVG itd. Nie martw się, jeśli nie wiesz, o co w nim chodzi. Dzięki tej książce już wkrótce większość tego niezrozumiałego kodu stanie się całkiem jasna.

OSTRZEŻENIE

Musisz pamiętać, że co prawda uczenie się w oparciu o czyjąś pracę jest w porządku, ale kradzież czyjegoś kodu jest złym pomysłem (lub wręcz nielegalnym). Jeśli chce się wykorzystać znaleziony kod, należy zawsze zapytać o zgodę jego autora i wymienić go jako twórcę.

Krótkie wprowadzenie do języka HTML

HTML zostanie dokładnie omówiony w drugiej części książki, dlatego nie chcę teraz wchodzić w szczegóły tego języka, jednak jest kilka kwestii dotyczących działania HTML-a i obsługiwanego przez przeglądarkę, które warto poruszyć już teraz.

Pora przejrzeć dokument HTML z rysunku 2.3 i porównać go z efektem widocznym w przeglądarce. Nietrudno zobaczyć, że elementy oznaczone znacznikami HTML w dokumencie źródłowym odpowiadają temu, co jest wyświetlane w oknie przeglądarki.

Po pierwsze, widać, że tekst znajdujący się wewnątrz nawiasów ostrych (na przykład `<body>`) nie jest wyświetlany w przeglądarce. Przeglądarka wyświetla tylko zawartość elementu, a znaczniki go otaczające są ukrywane. Znaczniki zawierają nazwy elementów HTML i zazwyczaj są skrótami od nazw angielskich, na przykład `h1` pochodzi od nagłówka pierwszego poziomu (ang. *heading level 1*), a `em` od zaakcentowanego tekstu (ang. *emphasized text*).

Po drugie, większość znaczników HTML występuje w parach otaczających zawartość elementu. W dokumencie HTML `<h1>` oznacza, że tekst następujący po nim powinien być nagłówkiem pierwszego poziomu, natomiast `</h1>` wskazuje na koniec nagłówka. Niektóre elementy, zwane **elementami pustymi**, nie mają zawartości. W przykładowym kodzie strony znacznik `<hr>` oznacza element pusty, który wskazuje przeglądarce, że w tym miejscu ma się znaleźć separator treści, który na ogół przyjmuje postać poziomej linii (zresztą nazwa elementu to skrót od *horizontal rule*).

Ponieważ kiedy zaczynałam moją przygodę z językiem HTML, nie miałam żadnego doświadczenia z programowaniem, pomogło mi myślenie o znacznikach oraz tekście jako o koralikach nawleczonych na nitkę, które przeglądarka „ściąga” jeden po drugim. Kiedy przeglądarka napotyka otwarty nawias (`<`), zakłada, że następujące po nim znaki są częścią znacznika — aż do nawiasu zamykającego (`>`). Podobnie zakłada, że cała treść znajdująca się po początkowym znaczniku `<h1>` będzie nagłówkiem — aż do napotkania znacznika końcowego `</h1>`. W ten sposób przeglądarka **przetwarza** dokument HTML. Zrozumienie sposobu działania przeglądarki przydaje się, kiedy rozwiązuje się problemy ze źle działającym dokumentem HTML.

Ale gdzie są obrazki?

W samym pliku HTML nie ma oczywiście obrazków. W jaki więc sposób znajdują się one w oknie przeglądarki? Na rysunku 2.3 widać, że każdy z obrazków jest osobnym plikiem graficznym. Grafiki są umieszczane w tekście za pomocą elementu `img` (ang. *image* — obraz) języka HTML, który instruuje przeglądarkę, gdzie ma znaleźć grafikę (zawiera jej adres URL). Kiedy przeglądarka napotyka element `img`, wykonuje kolejne żądanie do serwera w celu uzyskania pliku obrazka, a następnie umieszcza ten obrazek w treści strony.

Przeglądarka WWW wykonuje również żądania do serwera mające na celu pobranie arkuszy stylów (na przykład *kitchen.css*), skryptów JavaScript (*.js*) oraz osadzonych treści multimedialnych, na przykład dźwięku i wideo. Oprogramowanie przeglądarki WWW, a dokładniej: jej silnik, składa poszczególne elementy w jedną stronę.

Składanie strony odbywa się niemal natychmiast, dlatego wydaje się, że cała strona ładuje się razem. Przy wolnym połączeniu z internetem albo gdy strona zawiera duże obrazy lub media, proces składania strony może być wyraźnie widoczny, gdyż obrazki będą się pojawiały po wyświetleniu tekstu. Może się nawet okazać, że w miarę pobierania kolejnych obrazków, czcionek i arkuszy stylów przeglądarka musi na nowo wyświetlić stronę (choć można skonstruować stronę tak, by temu zapobiec).

Dodawanie stylów

Teraz chcę zwrócić uwagę na ostatni z kluczowych składników naszej prostej strony. Na początku dokumentu HTML znajduje się element `link`, który wskazuje plik arkusza stylów *kitchen.css*. Zawiera on kilka wierszy definiujących wygląd strony w przeglądarce. Składnia pliku odpowiada regułom ustalonym dla **kaskadowych arkuszy stylów** (ang. *Cascading Style Sheets* — CSS). Arkusze stylów służą do definiowania wyglądu (często mówi się o **prezentacji**) elementów znajdujących się w dokumencie HTML (czyli **struktury** dokumentu). W trzeciej części poznasz prawdziwą potęgę kaskadowych arkuszy stylów.

Rysunek 2.4 przedstawia stronę Kuchnia Kasi bez stylów oraz po ich zastosowaniu. Przeglądarki mają zdefiniowane domyślne style dla poszczególnych elementów HTML, które są stosowane, gdy w dokumencie nie ma odwołania do arkusza stylów (rysunek po lewej stronie). Czasem wystarczy dodać kilka reguł stylów, by całkowicie zmienić wygląd i charakter strony.

Dodawanie zachowań za pomocą JavaScriptu

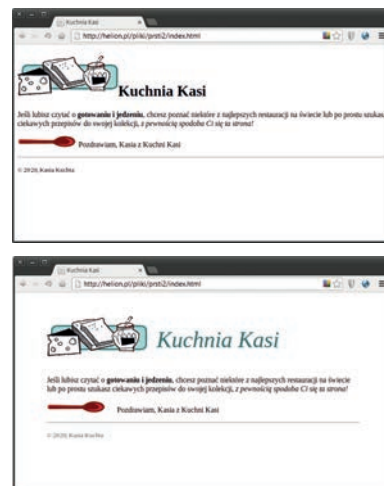
Aby elementy umieszczone na stronie *wykonywały* pewne operacje, konieczne jest użycie języka skryptowego o nazwie JavaScript (zapoznaj się z **uwagą** nieco dalej w rozdziale). Na omawianej tutaj stronie nie ma żadnych skryptów, ponieważ na początku książki chciałem zachować maksymalną prostotę przykładów. Warto jednak wiedzieć, że JavaScript to ważny składnik nowoczesnych witryn internetowych.

Podczas gdy HTML zapewnia strukturę, a CSS określa wygląd strony internetowej, JavaScript dodaje komponent zachowania kontrolujący sposób działania strony. Skrypt JavaScript może być oddzielnym plikiem na serwerze (wraz z rozszerzeniem *.js*) bądź też może być osadzony bezpośrednio w dokumencie HTML. Wykonanie skryptu może nastąpić natychmiast po wczytaniu strony lub następować na skutek pewnego działania ze strony użytkownika: kliknięcia myszą, umieszczenia kursora myszy nad elementem, wprowadzenia danych w polu formularza itd.

Krótkie wprowadzenie do języka JavaScript znajdziesz w czwartej części książki.

UWAGA

JavaScript nie jest wymagany do zapewnienia interaktywności łączny lub formularzy internetowych — do tych celów wystarcza sam kod HTML.



RYSUNEK 2.4. Strona Kuchnia Kasi przed zastosowaniem stylów (u góry) i po ich użyciu (u dołu)

Statyczna kontra dynamiczna witryna internetowa

Statyczna witryna internetowa składa się z plików HTML o niezmienniej treści, a każdy odwiedzający otrzymuje dokładnie te same informacje. Innymi słowy, każda strona wyświetlona przez przeglądarkę WWW jest odwzorowaniem jednego pliku HTML na serwerze. W książce skoncentrowałam się na tworzeniu statycznych stron, ponieważ są one całkiem proste i stanowią doskonały punkt wyjścia dla początkującego projektanta witryn internetowych.

Z kolei **dynamiczna witryna internetowa** jest generowana za pomocą kodu backendu, na przykład utworzonego w języku PHP lub ASP. W takim przypadku każda strona jest generowana przez aplikację w locie. W dynamicznej witrynie internetowej treść jest pobierana z bazy danych, a ostateczna postać strony może zostać dopasowana oddzielnie dla każdego użytkownika. Przygotowanie ogromnej dynamicznej witryny, składającej się z setek lub tysięcy stron, wymaga zdecydowanie mniej pracy niż utworzenie i przechowywanie każdej strony jako oddzielnego, statycznego dokumentu HTML.

Kody stanu HTTP

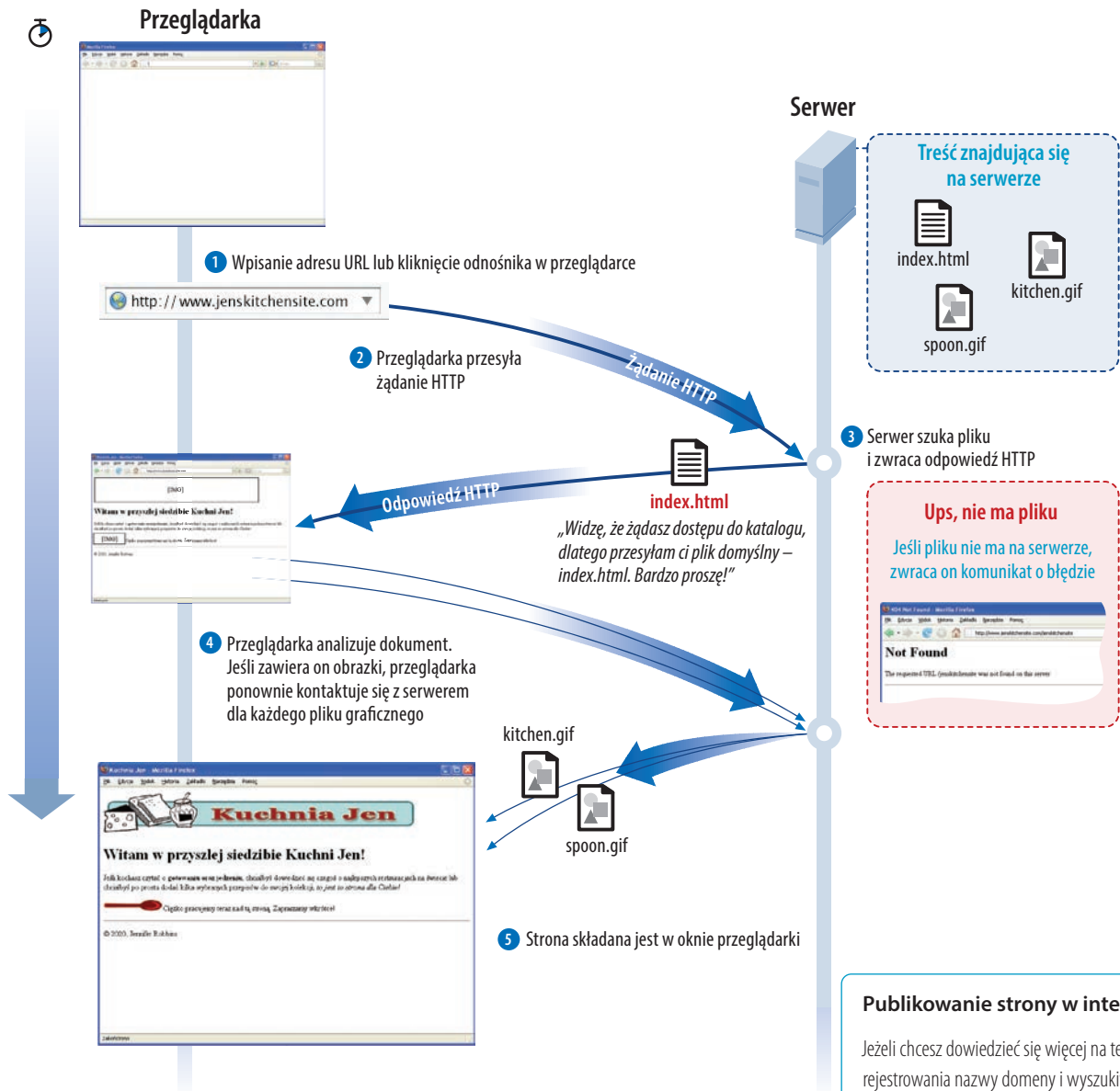
Serwerowy kod stanu jest generowany w odpowiedzi na żądanie wykonywane przez przeglądarkę WWW. Pełna lista kodów stanu jest całkiem długa (znajdziesz ją na stronie https://pl.wikipedia.org/wiki/Kod_odpowiedzi_HTTP), więc tutaj zdecydowałam się na podanie tylko kilku kodów, najczęściej generowanych w odpowiedzi na żądania:

- 200 — OK
- 301 — Trwale przeniesiony
- 302 — Znalezione
- 404 — Nie znaleziono
- 410 — Zniknął (nie jest dłużej dostępny)
- 500 — Wewnętrzny błąd serwera

SKŁADANIE WSZYSTKIEGO W CAŁOŚĆ

Aby podsumować nasze wprowadzenie do sposobu działania sieci WWW, warto prześledzić łańcuch zdarzeń, jaki występuje w przypadku każdej strony pojawiającej się w oknie przeglądarki (rysunek 2.5).

- 1 Żądanie strony odbywa się albo po wpisaniu jej adresu URL (na przykład `http://kuchniakasi.com`) bezpośrednio w przeglądarce, albo po kliknięciu odsyłacza znajdującego się na innej stronie internetowej. Adres URL zawiera wszystkie informacje potrzebne do odnalezienia dokumentu docelowego na określonym serwerze WWW. W omawianym przykładzie jest to plik domyślny (`index.html`) znajdujący się w katalogu głównym.
 - 2 Przeglądarka przesyła żądanie HTTP do serwera wymienionego w adresie URL i prosi o określony plik. Żądanie zawiera także informacje o językach znanych użytkownikowi oraz typach plików obsługiwanych przez przeglądarkę WWW. Jeśli URL jest adresem katalogu (a nie konkretnego pliku), proces jest taki sam, jakby żądało się pliku domyślnego z tego katalogu.
 - 3 Serwer szuka żadanego pliku i zwraca odpowiedź HTTP w postaci **nagłówka HTTP**. Ten nagłówek zawiera informacje o pliku, m.in.: datę ostatniej modyfikacji, wielkość pliku i jego wartość Content-Type, na przykład dla pliku `.html` wartością Content-Type jest `text/html`.
 - a. Jeśli strona nie może zostać odnaleziona, serwer zwraca komunikat o błędzie. Zazwyczaj brzmi on: „404 Nie znaleziono pliku” („404 Not found”), choć czasem mogą się pojawić bardziej przyjazne komunikaty. Istnieje możliwość wygenerowania innych typów błędu — zapoznaj się z ramką „Kody stanu HTTP”.
 - b. Jeśli dokument *zostaje* odnaleziony, serwer odczytuje go i zwraca do przeglądarki. W przypadku dynamicznej witryny internetowej serwer generuje stronę na podstawie otrzymanych danych, a dopiero później zwraca ją przeglądarce WWW.
 - 4 Przeglądarka analizuje dokument HTML. Jeśli strona zawiera obrazki (wskazywane przez elementy `img`) lub zewnętrzne zasoby, takie jak skrypty i arkusze stylów, przeglądarka ponownie kontaktuje się z serwerem w celu zażądania każdego pliku podanego w kodzie HTML.
 - 5 Przeglądarka wstawia każdy z obrazków do dokumentu w miejsce podane w elemencie `img`, stosuje arkusze stylów i wykonuje skrypty. *Voilà!* Użytkownik może już cieszyć się złożoną w całość stroną internetową.
- Warto w tym miejscu dodać, że przedstawiłam tradycyjny, prosty scenariusz składania w całość strony internetowej. Obecnie bardzo często zdarza się generowanie stron na podstawie systemów zarządzania treścią (CMS), przechowujących informacje w bazach danych i używających szablonów do dołączania w locie danych do strony. W takim przypadku krok 3b będzie miał postać znacznie bardziej skomplikowanego procesu przygotowania strony na podstawie różnych danych, a nie jedynie pobrania i obsłużenia istniejącego pliku.



RYSUNEK 2.5. Sposób wyświetlania strony internetowej przez przeglądarkę

SPRAWDŹ SIĘ!

Zagrajmy w grę „Rozszyfruj skrót”! Poniżej wypisaliśmy kilka podstawowych pojęć wspomnianych w tym rozdziale. Odpowiedzi znajdziesz w dodatku A.

1. HTML _____ a. Miejsce narodzin programu Mosaic, pierwszej przeglądarki graficznej.
2. W3C _____ b. Lokalizacja dokumentu lub zasobu internetowego.
3. CERN _____ c. Język znaczników wykorzystywany we wszystkich dokumentach internetowych.
4. CSS _____ d. Dopasowuje nazwy domen do liczbowych adresów IP.
5. HTTP _____ e. Protokół przesyłania plików.
6. IP _____ f. Protokół służący do przesyłania dokumentów w internecie.
7. URL _____ g. Język umożliwiający definiowanie wyglądu dokumentów HTML.
8. NCSA _____ h. Laboratorium fizyki cząstek, w którym narodziła się sieć WWW.
9. DNS _____ i. Internet Protocol.
10. FTP _____ j. Organizacja monitorująca technologie internetowe.

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 

Dobry webmaster? Inżynier i artysta w jednym!

Przybory narzędziowy webmastera zmienia się bardzo szybko. Technologie, które parę lat temu były ciekawym eksperymentem, dziś są powszechnie stosowane. Responsywność stron internetowych stała się oczywistością. Modele Flexbox i Grid zrewolucjonizowały tworzenie układów stron internetowych. Zmieniły się używane formaty grafik, wzrosła też znacząco liczba narzędzi, które należałoby poznać. Naturalnie, wciąż można wskazać aspekty projektowania, które pozostały niezmiennie – przejrzystość, funkcjonalność, estetykę, a także przestrzeganie standardów.

To kolejne, rozbudowane i uaktualnione wydanie świetnego podręcznika projektowania i programowania stron internetowych, zawierające wiedzę niezbędną do rozpoczęcia samodzielnej pracy. Opisano tu zasady tworzenia witryn WWW oraz przedstawiono język HTML i techniki CSS. Sporo miejsca poświęcono multimediom: grafice, plikom dźwiękowym i wideo. Znalazł się tutaj także znakomity wstęp do języka JavaScript, łącznie z jego bibliotekami, nakładkami i skryptami DOM. Warto też zwrócić uwagę na opis przydatnych narzędzi, takich jak procesory CSS, wykrywanie funkcjonalności, powłoka, menedżer zadań i system kontroli Git – nieodzownych każdemu, kto pisze nowoczesne witryny internetowe. Treść książki została uzupełniona licznymi przykładami, ćwiczeniami i bogatym materiałem dodatkowym.

NAJWAŻNIEJSZE ZAGADNIENIA:

- tworzenie stron za pomocą HTML i CSS, w tym modele CSS Flexbox i siatki
- korzystanie z JavaScriptu
- grafika: tworzenie, optymalizacja i możliwości formatu SVG
- tworzenie stron przeznaczonych zarówno na monitory, jak i na urządzenia mobilne
- projektowanie responsywne
- nowoczesne narzędzia projektanta stron WWW

JENNIFER NIEDERST ROBBINS

od ponad ćwierćwiecza projektuje strony internetowe. W 1993 roku zaprojektowała pierwszą komercyjną witrynę internetową – Global Network Navigator (GNN) dla wydawnictwa O'Reilly. Jest założycielką i organizatorką Artifact Conference poświęconej kwestiom związanym z projektowaniem witryn internetowych dla urządzeń mobilnych. Występowała na licznych konferencjach, a także wykladała na Johnson & Wales University w Providence w stanie Rhode Island.

Helion 	Sprawdź nasze szkolenia!		KOD KORZYŚCI Sięgnij po więcej! ▶	
 helion.pl	SZKOLENIA 			
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	AKADEMIA IT & BUSINESS HELIONSZKOLENIA.PL		ISBN 978-83-283-6203-1  9 788328 362031	
INFORMATYKA W NAJLEPSZYM WYDANIU			Cena: 129,00 zł	

O'REILLY®