

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Head First HTML with CSS & XHTML. Edycja polska

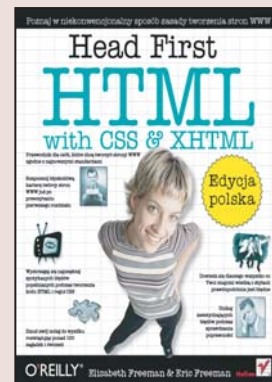
Autorzy: Eric Freeman, Elisabeth Freeman

Tłumaczenie: Piotr Rajca

ISBN: 83-246-0427-8

Tytuł oryginału: [Head First HTML with CSS & XHTML](#)

Format: 200×234, stron: 684



### Poznaj w niekonwencjonalny sposób zasady tworzenia stron WWW

- Wykorzystaj najnowsze standardy języka HTML
- Zastosuj style do zdefiniowania wyglądu strony
- Opracuj formularze i mechanizmy nawigacyjne

Język HTML stale ewoluuje. Od momentu swojego powstania przeszedł długą drogę. Początkowo był to zestaw kilku znaczników służących do formatowania tekstów naukowych, publikowanych w sieci będącej załącznikiem internetu. Obecnie technologia ta pozwala na niemal dowolne układanie elementów tekstowych i graficznych na stronach WWW wyświetlanych w przeglądarkach internetowych. Współczesny HTML to nie tylko znaczniki, ale także style – uniwersalne rozwiązanie umożliwiające kontrolowanie wyglądu strony. Zbudowanie nowoczesnej i szybko ładującej się witryny WWW wymaga opanowania tych elementów. Przeraża Cię to? Widzisz już w myślach setki linii przykładowego kodu, dziesiątki definicji i diagramów? Niepotrzebnie.

Dzięki książce „Head First HTML with CSS & XHTML. Edycja polska” poznasz najnowsze standardy tworzenia witryn WWW w sposób gwarantujący szybkie i bezstresowe przyswojenie wiedzy. Jej autorzy, wykorzystując najnowsze osiągnięcia teorii uczenia, przedstawiają wszystkie zagadnienia niezbędne do rozpoczęcia projektowania i tworzenia serwisów WWW z wykorzystaniem języka HTML oraz kaskadowych arkuszy stylów CSS. Poznasz znaczniki HTML, zasady umieszczania na stronach WWW elementów graficznych oraz sposoby formatowania tekstów za pomocą stylów. Jednak, co najważniejsze, nauczysz się stosować tę wiedzę w praktyce.

- Struktura dokumentu HTML
- Znaczniki formatujące
- Łącza do innych stron WWW
- Wstawianie elementów graficznych
- Zgodność ze standardami
- Formatowanie za pomocą stylów CSS
- Tworzenie formularzy

Przekonaj się, że nawet przy poznawaniu skomplikowanych technologii można się świetnie bawić.



## Spis treści (skrótowy)

Wprowadzenie	21
1 Język Sieci. <i>Poznajemy HTML</i>	33
2 Poznajemy „HT” z nazwy języka HTML. <i>Dokładniejsze poznanie hipertekstu</i>	73
3 Konstrukcja stron WWW. <i>Elementy konstrukcyjne</i>	107
4 Wycieczka do Webowic. <i>Nawiązujemy połączenie</i>	153
5 Poznajemy media. <i>Umieszczanie obrazów na stronach WWW</i>	193
6 Poważny HTML. <i>Standardy, zgodność z nimi i cały ten kram</i>	249
7 Dodanie „X” do HTML. <i>Zaczynamy stosować XHTML</i>	291
8 Delikatna stylizacja. <i>Poznajemy CSS</i>	311
9 Poszerzamy swoje słownictwo. <i>Style czcionek i kolorów</i>	365
10 Bliskie kontakty z elementami. <i>Model blokowy</i>	407
11 Zaawansowane sposoby konstruowania stron WWW. <i>Elementy div i span</i>	449
12 Rozmieszczanie elementów. <i>Układ i pozycjonowanie</i>	507
13 Mania tabel. <i>Tabele i listy</i>	567
14 W stronę interaktywności. <i>Formularze XHTML</i>	607
Dodatek. Dziesięć najważniejszych zagadnień (których nie opisaliśmy)	655

## Spis treści (na serio)

### Wprowadzenie

**Twój mózg koncentruje się na kodzie HTML i CSS.** Podczas gdy Ty starasz się czegoś nauczyć, Twój mózg robi Ci przysługę i dba o to, abyś przez przypadek *nie zapamiętał* zdobywanych informacji. Twój mózg myśli sobie: „Lepiej zostawić trochę miejsca na bardziej istotne informacje, na przykład: jakich zwierząt unikać albo czy jeżdżenie na snowboardzie nago jest dobrym pomysłem”. A zatem, w jaki sposób możesz oszukać swój mózg i przekonać go, że Twoje życie zależy od znajomości HTML-a i CSS?

Dla kogo jest ta książka?	22
Wiemy, co sobie myśli Twój mózg	23
Metapoznanie	25
Zmuś swój mózg do posłuszeństwa	27
Zespół recenzentów	30
Podziękowania	31

## Poznajemy HTML

# 1

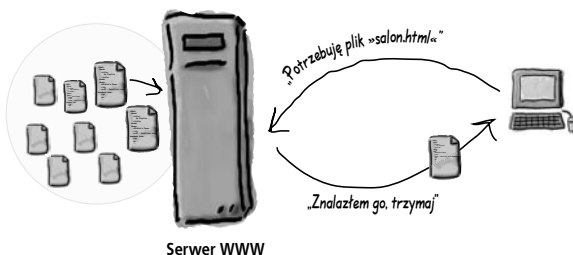
### Język Sieci

**Jedyną przeszkodą na drodze do zaznaczenia Twojej obecności w Sieci jest konieczność poznania nowego żargonu:** HTML-a (to skrót od słów: *HyperText Markup Language* — język znaczników hipertekstu). A zatem przygotuj się na kilka lekcji nowego języka. Po przeczytaniu tego rozdziału nie tylko będziesz rozróżniać niektóre proste **elementy** HTML, lecz także nabędziesz odpowiedniego **stylu**. W końcu, po przeczytaniu całej książki, będziesz posługiwać się tym językiem w taki sposób, jakbyś całe życie spędził w Webowicach.

Bez stresu, ale kiedy skończysz, naszą stronę będą odwiedzać tysiące osób. Dlatego nie tylko musi ona być poprawna, lecz także koniecznie musi świetnie wyglądać!



WWW zabiła radio	34
Co robi serwer?	35
Oto co piszesz (kod HTML)...	36
Co tworzy przeglądarka	37
Twoja długa przerwa w kafeterii Świat Kaw	41
Tworzenie strony kafeterii	43
Tworzenie plików HTML (Widnows)	44
W międzyczasie w kafeterii Świat Kaw	47
Otwieranie strony w przeglądarce	49
Testowanie strony WWW	50
Znaczniki bez tajemnic	55
Poznajemy element style	59
Nadawanie stylu stronie kafeterii	60
Kto co robi?	62
Pogawędki przy kominku	64
Kluczowe zagadnienia	66
Rozwiązania ćwiczeń	69



## Dokładniejsze poznanie hipertekstu

# 2

### Poznajemy „HT” z nazwy języka HTML

**Czy ktoś powiedział „hipertekst”?** A co to takiego? Ech... to jedynie *podstawa całej* WWW. W rozdziale 1. przyjrzelśmy się nieco językowi HTML i przekonaliśmy się, że jest to język wykorzystujący znaczniki („ML” z nazwy HTML) do opisywania struktury stron WWW. Teraz zajmiemy się możliwościami związanymi z pierwszą częścią nazwy „HTML” — „HT”, czyli *hipertekstem*, który pozwoli nam uwolnić się od ograniczeń jednej strony i tworzyć połączenia z innymi stronami WWW. Jednocześnie poznamy nowy, bardzo potężny element — `<a>` — i dowiemy się, że bycie „względny” jest całkiem fajne. A zatem zapnij pasy — właśnie zaczynamy poznawać hipertekst.



Salon Head First — nowy i poprawiony	74
Tworzenie nowej strony salonu	76
Co zrobiliśmy?	78
Co robi przeglądarka?	79
Atrybuty	83
Problemy techniczne	88
Planowanie ścieżek	90
Poprawianie błędnych adresów obrazów...	96
Rozwiązania ćwiczeń	102

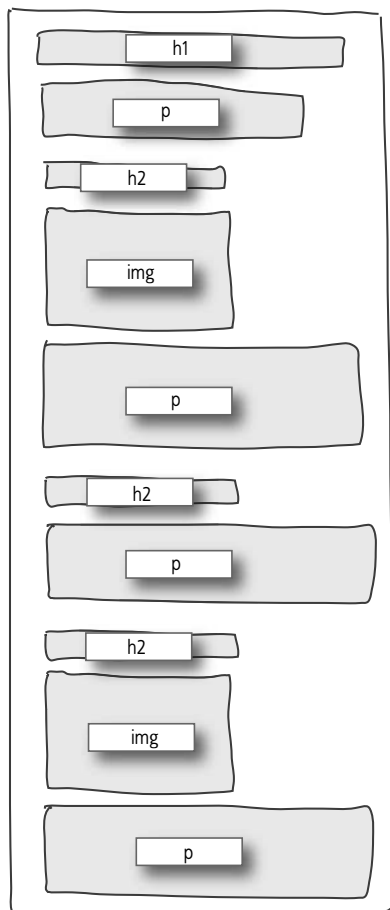


## Elementy konstrukcyjne

## 3

**Konstrukcja stron WWW****Czy powiedziano mi, że ta książka dotyczy tworzenia stron WWW?**

Bez wątpienia nauczyłeś się już całkiem sporo; poznałeś: znaczniki, elementy, łącza, ścieżki. Niemniej jednak cała ta wiedza będzie całkowicie nieprzydatna, jeśli nie wykorzystasz jej do tworzenia jakichś oszałamiających stron WWW. W tym rozdziale mamy zamiar przedstawić Ci proces tworzenia stron WWW: od pomysłu, poprzez projekt, stworzenie podstaw, wpisanie zawartości strony, aż do ostatecznych, drobnych modyfikacji związanych z dopracowywaniem jej wyglądu. Będzie Ci przy tym potrzebny jedynie kask oraz pas na narzędzia, gdyż pokażemy Ci nowe narzędzia, które bez wątpienia stałyby się chlubą każdego majsterkowicza.



Z dziennika na stronę WWW z prędkością 20 km na godzinę	109
Orientacyjny szkic projektu	110
Od szkicu do układu	111
Od układu do strony WWW	112
Testowanie strony WWW Antka	114
Poznajemy element <q>	116
Dłuuuuugie cytaty	120
Dodawanie elementu <blockquote>	121
Prawdziwe znaczenie zagadki związanej z elementami <blockquote> i <q>	124
Oczywiście można utworzyć listę, posługując się elementami <p>...	133
Tworzenie list HTML w dwóch prostych krokach	134
Umieszczanie jednego elementu wewnątrz innego nazywamy „zagnieżdżaniem”	139
Aby zrozumieć związki wynikające z zagnieżdżania elementów, narysuj odpowiedni szkic	140
Wykorzystanie zagnieżdżania dla zapewnienia poprawności znaczników elementu	141
Wewnątrzwierszowy czy blokowy?	143
Rozwiązania ćwiczeń	149

## Nawiązujemy połączenie

# 4

### Wycieczka do Webowic

#### Do serwowania stron WWW zdecydowanie najlepiej nadaje się Internet.

Do tej pory tworzyłeś strony WWW, które były przechowywane na Twoim lokalnym komputerze. Podobnie tworzone łącza także wskazywały na strony znajdujące się na Twoim komputerze. Jednak teraz zmienimy tę sytuację. W tym rozdziale spróbujemy Cię zachęcić, abyś umieścił swoje strony w Internecie, tak by Twoi przyjaciele, fani i klienci mogli je zobaczyć. Ujawnimy także tajemnice tworzenia łączy do innych stron, łącąc kod: h, t, t, p, :, /, /, w, w, w. A zatem spakuj swoje rzeczy — następnym przystankiem są Webowice.

Umieszczanie stron kafeiterii Świat Kaw (lub Twoich własnych) na WWW	154
Znajdowanie firmy hostingowej	155
W jaki sposób możesz zdobyć własną nazwę domeny?	156
Wprowadzamy się	158
Umieszczanie plików w katalogu głównym	159
Wszystkie informacje o FTP, jakie można zmieścić na dwóch stronach	160
Wracamy do pracy...	163
Ulica Główna, URL	164
Czym jest protokół HTTP?	165
Czym są ścieżki bezwzględne?	166
Sposób działania stron domyślnych	169
W jaki sposób tworzymy łącza do innych witryn WWW?	172
Łącza do strony o kofeinie	173
Dopracowanie i wykończenie strony	177
Łącza do konkretnych miejsc strony	179
Użycie elementu <a> do tworzenia punktów docelowych	180
Tworzenie łączy do odnośników	181
Wyświetlanie łączy w nowym oknie	185
Otwieranie nowych okien przy wykorzystaniu atrybutu target	186
Rozwiązania ćwiczeń	191

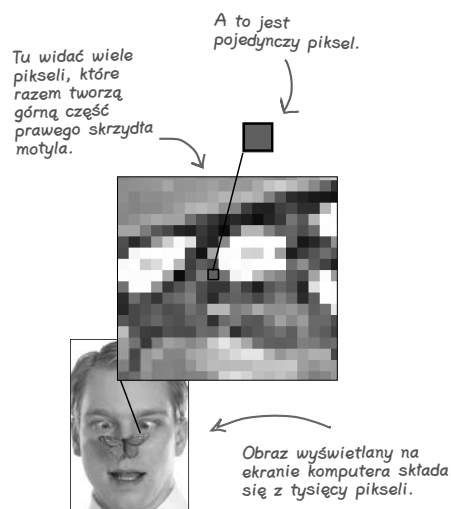


## Umieszczanie obrazów na stronach WWW

## 5

## Poznajemy media

**Teraz prosimy o uśmiech. A w zasadzie uśmiechnij się i powiedz: „gif”, „jpg” lub „png”** — bo to właśnie będą Twoje ulubione formaty graficzne, przynajmniej jeśli chodzi o robienie obrazów na potrzeby stron WWW. W tym rozdziale dowiesz się wszystkiego na temat umieszczania na stronach najczęściej spotykanego typu plików multimedialnych — obrazów. Czy masz jakieś zdjęcia cyfrowe, które musisz umieścić na swojej stronie? Nie ma problemu. Masz swoje logo, które też chciałbyś umieścić na stronie? Skonwertowałeś je do odpowiedniego formatu. Jednak zanim poznasz te wszystkie informacje, powinniśmy Ci chyba oficjalnie przedstawić element `<img>`, nieprawdaż? Zatem przepraszamy, zachowaliśmy się niegrzecznie, ale nigdy nie pokazano nam, jak należy się właściwie zachować w takiej sytuacji. Aby naprawić nasz błąd, zamieścimy w niniejszej książce cały rozdział poświęcony wyłącznie elementowi `<img>`. Gdy skończysz go czytać, będziesz znać wszelkie tajniki stosowania zarówno tego elementu, jak i wszystkich jego atrybutów. Dodatkowo zobaczysz, jak ten niewielki element potrafi zmusić przeglądarkę do wykonania dodatkowej pracy i wyświetlania obrazów.



Jak przeglądarki obsługują obrazy	194
Sposób działania obrazów	197
A teraz oficjalna prezentacja: Panie i Panowie, oto element <code>&lt;img&gt;</code>	201
Zawsze podawaj tekst alternatywny	204
Tworzenie świetnej rozrywkowej witryny: mój iPod	206
O rany! Obraz jest o wiele za duży	209
Poprawa kodu HTML stron „mój iPod”	219
Przystosowanie strony do wykorzystania miniatur	223
Zmiana obrazów w łącza	227
A zatem w jaki sposób mogę stworzyć łącza będące obrazami?	229
Jakiego formatu powinniśmy użyć?	234
Być przezroczystym czy nie być przezroczystym? Oto jest pytanie...	235
Chwila, ale jaki jest kolor tła strony?	237
Sprawdzenie logo po utworzeniu otoczki	238
Umieszczenie logo na stronie głównej witryny	239
Rozwiązania ćwiczeń	243

## Standardy, zgodność z nimi i cały ten kram

# 6

### Poważny HTML

**Co jeszcze należy wiedzieć o języku HTML?** Przebyłeś już całkiem spory odcinek drogi wiodącej do biegłego poznania języka HTML. Czy nie nadszedł już czas, by zająć się CSS i dowiedzieć się, jak sprawić, by cały ten mdły kod zmienił się w bajeczną stronę? Zanim to jednak zrobimy, musimy się upewnić, że Twój kod HTML jest dobry (no wiesz, wymuskany, dopracowany, perfekcyjny); a zrobimy to, podchodząc bardzo poważnie do jego postaci i sposobu, w jaki go piszesz. Nie zrozum nas źle — cały czas pisałeś doskonały kod HTML, niemniej jednak istnieje jeszcze kilka rzeczy, które możesz zrobić, aby pomóc przeglądarce w wyświetlaniu stron oraz aby zabezpieczyć się przed drobnymi błędami, które mogą wkraść się do Twego kodu. A po co to wszystko? Otóż chodzi o strony, które w bardziej jednolity sposób będą wyświetlane w wielu różnych przeglądarkach (a nawet będą poprawnie wyświetlane na urządzeniach przenośnych i odczytywane przez przeglądarki głosowe używane przez osoby mające problemy ze wzrokiem), strony, które będą szybciej wczytywane i które zapewniają poprawną współpracę ze stylami CSS. Przygotuj się, gdyż w tym rozdziale zmienisz się z HTML-owego majsterkowicza w prawdziwego profesjonalistę.



Krótką historią języka HTML	252
Nie możemy pozwolić, by przeglądarka pracowała w trybie „quirks”!	255
Dodawanie definicji typu dokumentu	257
Poznajemy analizator poprawności W3C	260
Weryfikacja stron salonu Head First	261
Houston, mamy problem...	262
Dodanie znacznika <meta> do określenia typu zawartości strony	264
Uszczęśliwienie analizatora W3C dzięki użyciu znacznika <meta>...	265
Houston, mamy kolejny problem...	267
Do trzech razy sztuka?	269
Zmiana definicji na ściśłą	273
Czy weryfikacja strony zakończyła się pomyślnie?	274
Rozwiązanie problemu z zagnieżdżaniem	276
Jeszcze jedna próba: czy jesteśmy dostatecznie ściśli...	277
Ściśła wersja HTML-a 4.01 — sięgnij po informator	279
Pogawędki przy kominku	283
Archeologia HTML	286
Rozwiązania ćwiczeń	289



## Zaczynamy stosować XHTML

## 7

## Dodanie „X” do HTML

**Mamy jeden straszny sekret, którego jeszcze Ci nie wyjawiliśmy.** Wiemy, że *myślałeś*, iż kupujesz książkę dotyczącą języka HTML, jednak tak naprawdę jest to książka o zamaskowanym języku XHTML. W rzeczywistości prawie cały czas uczyliśmy Cię właśnie XHTML-a. Zapewne zastanawiasz się, czym właściwie jest ten XHTML. Cóż, przedstawiamy zatem rozszerzalny HTML (ang. *eXtensible HTML*), który znany jest także jako XHTML — kolejne wcielenie języka HTML. Jest lepszy, fajniejszy, a nawet lepiej dostosowany do zachowania zgodności z przeglądarkami działającymi na bardzo różnych urządzeniach. W tym krótkim rozdziale mamy zamiar przeprowadzić Cię od HTML-a do XHTML-a w trzech prostych etapach. A zatem obróć stronę i już znacząco zbliżysz się do celu... (a tym samym także do CSS).

Czym jest XML?	293
Co to wszystko ma wspólnego z HTML-em?	294
Dlaczego więc mógłbyś chcieć używać XHTML-a?	296
Stosowanie XHTML-a 1.0 w wersji Strict	298
Trzy etapy przejścia ze ścisłej wersji języka HTML 4.01 do XHTML-a 1.0	300
Stara szkoła ścisłej wersji języka HTML 4.01	301
Nowy, poprawiony zapis stosowany w XHTML 1.0	301
Weryfikacja — nie tylko dla HTML-a	303
Pogawędki przy kominku	306
HTML czy XHTML? Oto jest pytanie...	308
Rozwiązania ćwiczeń	309

Lubię być na bieżąco z najnowszymi trendami i technologiami. XHTML to przyszłość, a ponieważ jest niezwykle podobny do HTML-a, to dlaczego nie miałabym skorzystać z tej nowszej i lepszej technologii?



# 8

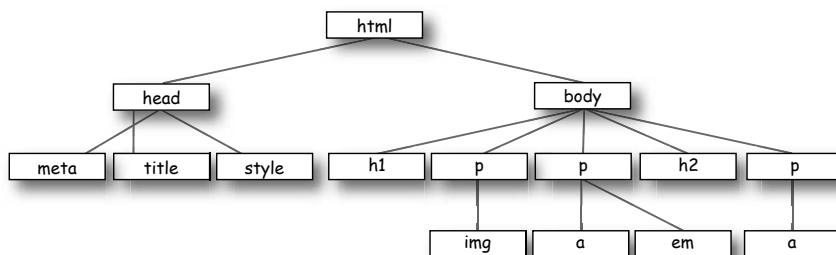
## Delikatna stylizacja

**Mówiono mi, że to będzie książka o CSS.** Jak do tej pory koncentrowałeś się na poznawaniu języka XHTML używanego do tworzenia struktury stron WWW. Jednak, jak sam się mogłeś przekonać, pojęcie stylu i gustu, jaki mają przeglądarki, pozostawia wiele do życzenia. Oczywiście moglibyśmy wezwać policję do spraw mody, ale nie musimy tego robić. Dzięki zastosowaniu CSS będziemy mieli całkowitą kontrolę nad sposobem prezentacji strony, a niejednokrotnie nie będziemy nawet musieli wprowadzać w tym celu jakichkolwiek zmian w naszym kodzie XHTML. Czy to naprawdę może być aż tak proste? Cóż, *będziesz* musiał nauczyć się zupełnie nowego języka; w końcu Webowice to miasto, w którym obowiązują dwa języki urzędowe. Po przeczytaniu zamieszczonego w tym rozdziale samouczka do nauki języka CSS będziesz mógł stanąć po *dowolnej* stronie ulicy Głównej i swobodnie prowadzić konwersację.

Zagadka  
na pięć  
minut



Już nie jesteś w Kansas...	312
Zasłyszane na targu w Webowicach	314
Stosowanie CSS z językiem XHTML	315
Dodatkowo wyświetlmy linię pod nagłówkiem strony	321
Określanie drugiej reguły, dotyczącej wyłącznie elementu <h1>	322
Dodanie salonowego stylu do strony o drinkach oraz sposobie dojazdu	329
Łączenie pliku <i>salon.html</i> z zewnętrznym arkuszem stylów	331
Czas porozmawiać o dziedziczeniu...	337
Co się stanie, jeśli przeniesiemy określenie czcionki do jakiegoś elementu na wyższym poziomie hierarchii?	338
Przesłanianie dziedziczenia	340
Tworzenie selektora klasy	344
Więcej informacji o selektorach klas	346
Najmniejszy i najszybszy przewodnik dotyczący stosowania stylów dostępny na świecie	348
Kto dziedziczy?	352
Zapewnienie poprawności kodu CSS salonu Head First	355
Rozwiązania ćwiczeń	362



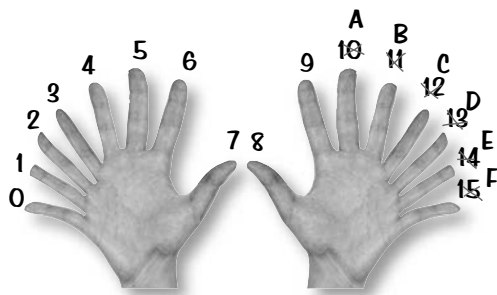
## Style czcionek i kolorów

## 9

**Poszerzamy swoje słownictwo**

**Twoje lekcje CSS przebiegają sprawnie i miło.** Opanowałeś już podstawy CSS, wiesz, jak tworzyć reguły i określać style elementów. Nadszedł czas, abyś rozszerzył swoje słownictwo, a to oznacza poznanie nowych właściwości i dowiedzenie się, do czego można ich użyć. W tym rozdziale poznasz niektóre spośród najczęściej używanych właściwości określających sposób prezentacji tekstu. Do tego jednak będzie Ci potrzebna znajomość kilku zagadnień związanych z czcionkami i kolorami. Dowiesz się, że Twoje możliwości nie ograniczają się do czcionek, które wszyscy powszechnie wykorzystują, ani domyślnych wielkości i kolorów znaków, jakie przeglądarki używają do prezentowania tekstu akapitów i nagłówków. Oprócz tego przekonasz się, że o kolorach można się dowiedzieć znacznie więcej, niż mogłoby się zdawać na pierwszy rzut oka.

Tekst i czcionki z wysokości 10 km	366
Ale czym tak w zasadzie są „rodziny czcionek”?	368
Określanie rodziny czcionek w CSS	371
Odkurzamy dziennik Antka	372
W jaki sposób radzić sobie z faktem, że każdy może mieć inne czcionki?	375
Czyli w jaki sposób powinienem określać wielkości czcionek?	378
Wprowadźmy zatem zmiany wielkości czcionek na stronie dziennika Antka	380
Modyfikacja grubości czcionki	383
Nadawanie czcionkom odrobiny stylu	385
Zmiana postaci cytatów poprzez zastosowanie kursywy	386
W jaki sposób tworzone są kolory na stronach WWW?	388
W jaki sposób można określać kolory na stronach WWW? Ile jest tych sposobów...	391
Dwuminutowy przewodnik po kodach szesnastkowych	394
W jaki sposób można określać wartości kolorów?	396
Wróćmy do strony Antka...	399
Wszystko, co chciałbyś wiedzieć o właściwości text-decoration	401
Usuwanie podkreślenia...	402
Rozwiązania ćwiczeń	404



## Model blokowy

# 10

## Bliskie kontakty z elementami

**Aby konstruować strony WWW na zaawansowanym poziomie, musisz bardzo dobrze poznać materiały konstrukcyjne, jakimi dysponujesz.** W tym rozdziale dokładnie przyjrzymy się naszym materiałom konstrukcyjnym, czyli elementom XHTML. Dokładnie zbadamy elementy blokowe oraz wewnątrzwierszowe pod mikroskopem i sprawdzimy, z czego się one składają. Przekonasz się także, w jaki sposób przy wykorzystaniu CSS można kontrolować niemal wszystkie aspekty wyglądu elementów. Ale to jeszcze nie wszystko — dowiesz się także, w jaki sposób można przydzielać elementom unikalną tożsamość. A jeśli i to Ci nie wystarczy, to dowiesz się, w jakich sytuacjach i dlaczego mógłbyś używać kilku różnych arkuszy stylów. Obróć więc stronę i zacznij poznawać najskrytsze tajemnice elementów.

Aktualizacja strony salonu Head First	408
Zacznijmy od kilku prostych zmian	410
Sprawdzanie nowej wysokości wierszy	412
Przygotowania do gruntownych modyfikacji	413
Dokładniejsza analiza modelu blokowego	414
Jak można konfigurować bloki elementów?	416
Tworzenie stylu określającego postać akapitu gwarancji	421
Wypełnienie, obramowanie oraz marginesy akapitu gwarancji	423
Dodanie wypełnienia	423
A teraz dodajmy marginesy	424
Dodawanie obrazu tła	426
Poprawa obrazu tła	429
Jak dodać wypełnienie z lewej strony elementu?	430
Jak powiększyć margines z prawej strony elementu?	431
Dwuminutowy przewodnik po obramowaniach	432
Określenie postaci obramowania i zakończenie prac	434
Wywiad z klasą HTML	436
Atrybut id	438
Zastosowanie identyfikatora na stronie salonu	440
Miksowanie arkuszy stylów	442
Stosowanie wielu różnych arkuszy stylów	443
Rozwiązania ćwiczeń	448



## 11

Elementy `div` i `span`

## Zaawansowane sposoby konstruowania stron WWW

**Czas przygotować się do tworzenia poważnych konstrukcji.** W tym rozdziale przedstawimy dwa nowe elementy języka XHTML — `<div>` oraz `<span>`. Jednak nie są to zwykłe „kątowniki”, to „stalowe dźwigary” z prawdziwego zdarzenia. Posługując się tymi dwoma elementami, będziesz tworzył poważne konstrukcje wspomagające, a kiedy już je umieścisz na stronie, to będziesz mógł określać ich postać przy użyciu stylów na wiele sposobów dających bardzo duże możliwości. Wcześniej nie mogliśmy raczej nic wymyślić, jednak zauważ, że Twój pas na narzędzia CSS powoli zaczyna się wypełniać, a zatem nadszedł czas, by pokazać nowe rozwiązania, które znacznie uprościć określanie tych wszystkich właściwości. Oprócz tego zaprosiliśmy do niniejszego rozdziału kilkoro ciekawych gości, tak zwane *pseudoklasy*, które pozwolą Ci na tworzenie bardzo interesujących selektorów. (Jeśli uważasz, że słowo „pseudoklasa” mogłoby być doskonałą nazwą dla Twojej następnej kapeli, to wiedz, że się spóźniłeś — wyprzedziliśmy Cię).



Dokładniejsza analiza kodu XHTML listy napojów	451
Zobaczmy, jak można podzielić stronę na logiczne sekcje	453
Dodawanie obramowania	460
Sprawdzenie obramowania	460
Dodawanie faktycznych stylów do sekcji napojów tygodnia	461
Plan gry	462
Określanie szerokości sekcji napojów	462
Dodawanie podstawowych stylów do sekcji napojów	467
Potrzebujemy sposobu wybierania elementów podrzędnych	473
Zmiana kolorów nagłówków w sekcji napojów	475
Zmiana wysokości wierszy	476
Czas, by pójść na skróty	478
Trzy proste kroki dodawania elementów <code>&lt;span&gt;</code>	484
Element <code>&lt;a&gt;</code> oraz wszystkie jego osobowości	488
W jaki sposób można określać styl elementów na podstawie ich stanu?	489
Stosowanie pseudoklas	491
Czy wzmianka o „kaskadzie” nie byłaby na czasie?	493
Kaskada	495
Zapraszamy do gry: „Jaki jest mój poziom precyzji?”	496
Składanie wszystkich elementów w jedną całość	497
Rozwiązania ćwiczeń	504

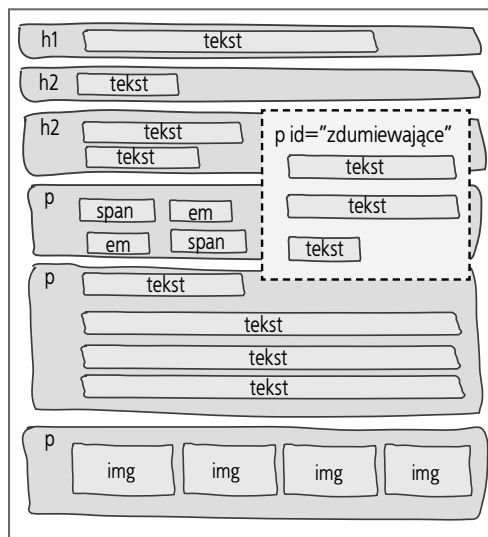
# 12

## Układ i pozycjonowanie

### Rozmieszczanie elementów



**Czas nauczyć Twoje elementy XHTML nowych sztuczek.** Już ani chwili dłużej nie pozwolimy elementom XHTML siedzieć na miejscu i lenić się — nadszedł czas, by w końcu się ruszyły i pomogły nam w tworzeniu stron, które mają jakiś prawdziwy *układ*. Ale jak? No cóż... już dosyć dobrze poznałeś strukturalne elementy `<div>` oraz `<span>` i wiesz już wszystko na temat modelu blokowego, nieprawdaż? Nadszedł zatem czas, by wykorzystać tę całą wiedzę do tworzenia układów stron. Nie obawiaj się... nie będziemy już więcej pisać o tłach i kolorach czcionek, skoncentrujemy się na tworzeniu profesjonalnych, wielokolumnowych układów stron. W tym rozdziale cała wiedza, jaką do tej pory zdobyłeś, zacznie się układać w jedną logiczną całość.



Czy wykonałeś ćwiczenie „Supermoc umysłu”?	508
Użyj rozkładu, Luke...	509
A co z elementami wewnątrzwierszowymi?	511
Jak to wszystko działa?	512
Jak stworzyć „pływający” element?	515
Za kulisami na stronie salonu	517
Nowa wersja strony kafeiterii Świat Kaw	519
Przenieś pasek boczny tuż poniżej nagłówka strony	524
Określ szerokość paska i zmień go na element pływający	524
Rozwiązanie problemu dwóch kolumn	527
Określanie szerokości marginesu głównej sekcji strony	528
Wracamy do rozwiązania problemu przesłaniających się elementów	531
Z prawej mocno, z lewej swobodnie...	534
Układy elastyczne i stałe	537
Jak działa pozycjonowanie bezwzględne	540
Stosowanie pozycjonowania bezwzględnego	543
Dobry kompromis pozwala na rozwiązanie problemu stopki	547
Pozycjonowanie obrazu pucharu	549
Jak działa pozycjonowanie ustalone?	555
Zastosowanie ujemnej wartości właściwości left	557
Pozycjonowanie względne	559
Dążymy w kierunku trzech kolumn, a może nawet dalej...	561
Rozwiązania ćwiczeń	564

## Tabele i listy

## 13

## Mania tabel

**Jeśli to wygląda jak tabela i mówi jak tabela...** Wcześniej czy później nadejdzie ten moment, kiedy człowiek musi się zmierzyć z *przerazającymi danymi tabelarycznymi*. Niezależnie od tego, czy będzie to strona przedstawiająca zapasy magazynowe firmy, czy też najnowszy katalog chińskich podróbek lalki Barbie (nie obawiaj się, nie zadenuncjujemy Cię), to musisz być świadom, iż trzeba ją sporządzić w XHTML-u. Ale jak to zrobić? Cóż, mamy dla Ciebie propozycję: złóż zamówienie już teraz, a my w jednym rozdziale ujawnimy wszystkie sekrety, które pozwolą Ci umieścić Twoje dane bezpośrednio w ślicznej tabeli XHTML. Ale to jeszcze nie wszystko: do każdego zamówienia dodajemy unikalny, a co ważniejsze, darmowy poradnik dotyczący określania postaci tabel XHTML przy wykorzystaniu kaskadowych arkuszy stylów. A jeśli zamówienie złożysz natychmiast, to w ramach specjalnej nagrody otrzymasz podobny poradnik dotyczący określania postaci list XHTML. A zatem nie wahaj się ani chwili! Zadzwoń już teraz!

Jak tworzy się tabele w języku XHTML?	569
Tworzenie tabeli przy użyciu języka XHTML	570
Jak to wyświetla przeglądarka	571
Analiza tabel	572
Dodawanie tytułu i podsumowania	575
Zanim zaczniemy określać style, umieścimy kod tabeli na stronie Antka	577
Łączenie podwójnych obramowań komórek	582
A może by tak dodać trochę koloru?	584
Antek dokonał bardzo interesującego odkrycia...	585
Kolejny rzut oka na tabelę Antka	586
Jak stworzyć komórkę rozciągającą się na kilka wierszy?	587
Nowa, poprawiona tabela	589
Kłopoty w rajcu?	590
Przesłanianie stylów dla nagłówków zagnieżdżonej tabeli	594
Ostatnie poprawki strony Antka	595
Rozwiązania ćwiczeń	600



Miejscowość	Data	Temperatura	Wysokość	Populacja	Ocena obiadu
Walla Walla (Waszyngton)	15 czerwca	24	360 m n.p.m.	29 686	4/5
Magic City (Idaho)	25 czerwca	23	1 594 m n.p.m.	50	3/5
Bountiful (Utah)	10 lipca	33	1 268 m n.p.m.	41 173	4/5
Last Chance (Kolorado)	23 lipca	39	1 434 m n.p.m.	256	3/5
Truth or Consequences (Nowy Meksyk)	9 sierpnia	34	1 273 m n.p.m.	7 289	5/5
	27 sierpnia	35			Antek 4/5 Tereska 5/5
Why (Arizona)	18 sierpnia	40	258 m n.p.m.	480	3/5

## 14

## W stronę interaktywności

Jak do tej pory cała komunikacja pomiędzy tworzonymi witrynami a osobami, które je odwiedzają, odbywała się w jednym kierunku: z serwera do użytkownika. Jejku, czy nie byłoby fajnie, gdyby użytkownik też mógł nam coś powiedzieć? Właśnie taką możliwość dają nam formularze XHTML: kiedy już umieścisz je na stronach WWW (dodatkowo będzie potrzebna także pomoc ze strony serwera), to zapewnią one możliwość zbierania opinii od użytkowników oglądających witrynę rejestrowania zamówień przez Internet, wykonania kolejnego ruchu w internetowej grze czy też gromadzenia głosów w ankiecie „czaderski czy lamerski?”. W tym rozdziale poznasz całą drużynę elementów XHTML, które służą do tworzenia formularzy. Oprócz tego dowiesz się nieco o tym, co w niewidoczny dla użytkownika sposób musi robić serwer, by obsługiwać formularze. W końcu poświęcimy także nieco uwagi określaniu postaci formularzy przy użyciu stylów (choć jest to nieco kontrowersyjne zagadnienie; przeczytaj, a dowiesz się dlaczego).



Jak działają formularze?	608
Sposób działania formularzy w przeglądarce	609
Jaki jest kod XHTML formularza?	610
Co tworzy przeglądarka	611
Jak działa element <form>?	612
Przygotowania do stworzenia formularza Ziarnotronu	620
Dodawanie elementu <form>	621
Jak działają nazwy elementów formularzy?	622
Wróćmy do dodawania elementów <input> do kodu XHTML strony	624
Dodawanie kolejnych pól do formularza	625
Dodawanie elementu <select>	626
Dajemy klientom możliwość wyboru postaci dostarczonej kawy	628
Naciskanie przycisków opcji	629
Dokończanie formularza	630
Dodawanie pól wyboru oraz obszaru tekstowego	631
Przedstawienie działania metody GET	637
Z tabelami czy bez tabel? Oto jest pytanie...	642
Umieszczanie elementów formularza w tabeli	643
Dopracowanie wyglądu formularza i tabel przy użyciu stylów	646
Rozwiązania ćwiczeń	652



## 15

Dodatek. Pozostałości

**Dziesięć najważniejszych zagadnień (których nie opisaliśmy)**

**Opisaliśmy bardzo wiele zagadnień i już niemal dotarliśmy do końca niniejszej książki.** Będziemy za Tobą tęsknić, jednak zanim pozwolimy Ci odejść, chcielibyśmy mieć poczucie, że nie wypuszczamy Cię na szerokie wody WWW bez dodatkowego przygotowania. Bez wątplenia nie możemy umieścić w tym, stosunkowo niewielkim, rozdziale wszystkiego, co mogłoby Ci się przydać. Tak naprawdę to początkowo *umieściliśmy* tu wszystko, co powinieneś wiedzieć o językach XHTML i CSS (i o czym nie pisaliśmy w poprzednich rozdziałach), jednak musieliśmy zmniejszyć czcionkę do wielkości 0,00004 punktu. Wszystko się zmieściło, ale nikt nie byłby tego w stanie przeczytać. Dlatego odrzuciliśmy większość z tych zagadnień i pozostawiliśmy tu jedynie dziesięć najważniejszych.



Więcej informacji o selektorach	656
Ramki	658
Multimedia i Flash	659
Narzędzia do tworzenia stron WWW	660
Skrypty wykonywane w przeglądarce	661
Skrypty wykonywane na serwerze	662
Modyfikacje pod kątem mechanizmów wyszukiwawczych	663
Więcej informacji na temat arkuszy stylów używanych do drukowania stron	665
Strony przeznaczone dla urządzeń przenośnych	665
Blogi	666

**Skorowidz**

667

## Poznajemy „HT” z nazwy języka HTML



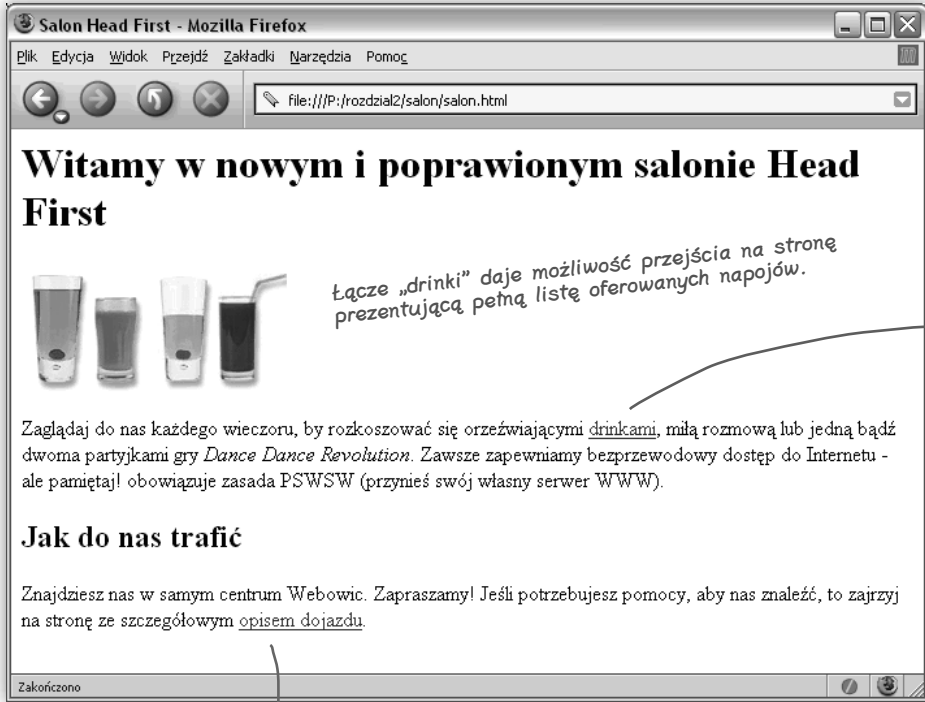
**Czy ktoś powiedział „hipertekst”?** A co to takiego? Ech...to jedynie *podstawa całej* WWW. W rozdziale 1. przyjrzeliśmy się nieco językowi HTML i przekonaliśmy się, że jest to język wykorzystujący znaczniki („ML” z nazwy HTML) do opisywania struktury stron WWW. Teraz zajmiemy się możliwościami związanymi z pozostałą częścią nazwy HTML-a — „HT” — czyli *hipertekstem*, który pozwoli nam uwolnić się z ograniczeń jednej strony i tworzyć połączenia z innymi stronami WWW. Jednocześnie poznamy nowy, bardzo potężny element — `<a>` — i dowiemy się, że bycie „względny” jest całkiem fajne. A zatem zapnij pasy — właśnie zaczynamy poznawać hipertekst.

## Salon Head First — nowy i poprawiony

Czy pamiętasz jeszcze stronę salonu Head First? Świetna strona, ale czy nie byłoby jeszcze lepiej, gdyby użytkownicy mogli przejrzeć listę serwowanych napojów? Co więcej, powinniśmy zamieścić informacje o sposobie dojazdu, tak by klienci mogli do nas trafić.

Oto nowa i poprawiona strona.

Dodaliśmy łącza do dwóch nowych stron — jednej z informacjami o napojach, a drugiej z opisem dojazdu.



Łącze „drinki” daje możliwość przejścia na stronę prezentującą pełną listę oferowanych napojów.

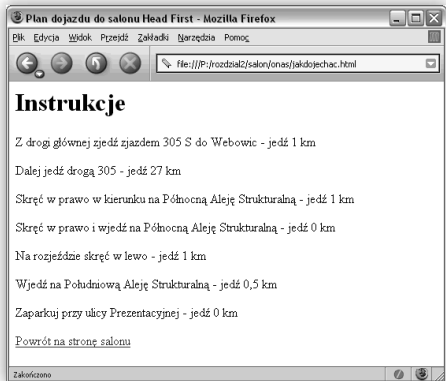
Zagładaj do nas każdego wieczoru, by rozkoszować się orzeźwiającymi drinkami, miłą rozmową lub jedną bądź dwoma partyjkami gry *Dance Dance Revolution*. Zawsze zapewniamy bezprzewodowy dostęp do Internetu - ale pamiętaj! obowiązuje zasada PSWSW (przynies swój własny serwer WWW).

### Jak do nas trafić

Znajdziesz nas w samym centrum Webowic. Zapraszamy! Jeśli potrzebujesz pomocy, aby nas znaleźć, to zajrzyj na stronę ze szczegółowym opisem dojazdu.

Zakończono

Łącze z „opisem dojazdu” prowadzi na stronę zawierającą szczegółowy opis dojazdu.



Plan dojazdu do salonu Head First - Mozilla Firefox

Plk Edycja Wdok Przejd Zakładki Narzędzia Pomoc

file:///P:/rozdzial2/salon/onas/jakdojehac.html

### Instrukcje

Z drogi głównej zjedź zjazdem 305 S do Webowic - jedź 1 km

Dalej jedź drogą 305 - jedź 27 km

Skręć w prawo w kierunku na Północną Aleję Strukturalną - jedź 1 km

Skręć w prawo i wjeźdź na Północną Aleję Strukturalną - jedź 0 km

Na rozjeździe skróć w lewo - jedź 1 km

Wjeźdź na Południową Aleję Strukturalną - jedź 0,5 km

Zaparkuj przy ulicy Prezentacyjnej - jedź 0 km

[Powrót na stronę salonu](#)

Zakończono

jakdojehac.html



drinki.html

Strona zawiera listę orzeźwiających i zdrowych napojów. Może się skusisz na jednego przed dalszą lekturą?

### Tworzenie nowej, poprawionej strony salonu Head First składa się z trzech etapów...

Spróbujmy przerobić oryginalną stronę salonu Head First i umieścić na niej łącza do dwóch nowych stron.

❶ Pierwszy etap jest łatwy, gdyż sami stworzyliśmy dla Ciebie pliki *jakdojehac.html* oraz *drinki.html*. Znajdziesz je w przykładach do książki, w witrynie [ftp.helion.pl/przyklady/hfhtcs.zip](http://ftp.helion.pl/przyklady/hfhtcs.zip).



❷ W ramach drugiego etapu zmodyfikujesz plik *salon.html*, dodając do niego łącza do plików *jakdojehac.html* oraz *drinki.html*.

❸ W końcu sprawdzisz strony, aby przekonać się, czy łącza działają zgodnie z oczekiwaniami. Kiedy skończysz, przyjrzymy się dokładnie, jak to wszystko działa.

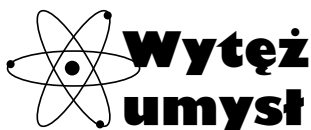
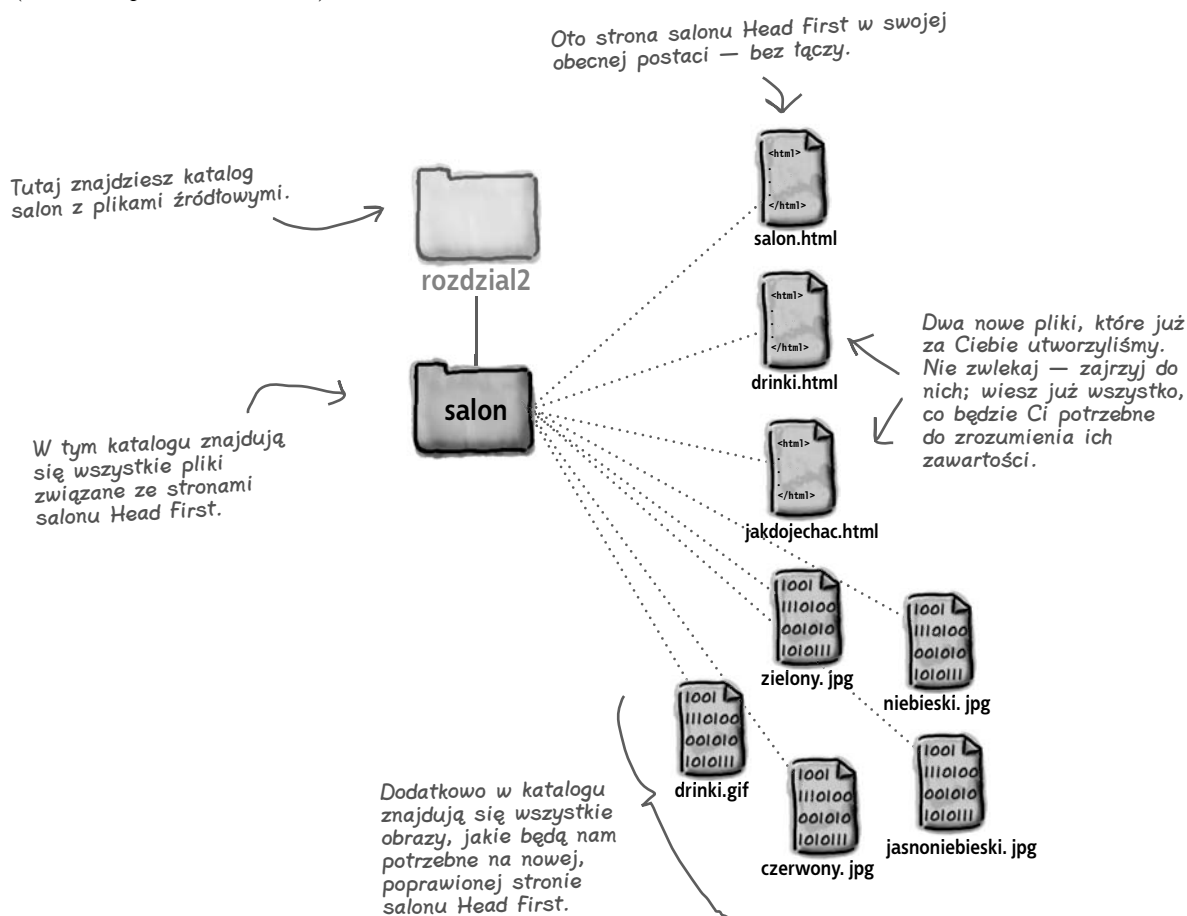
A teraz przewróć stronę i zaczynamy...

## Tworzenie nowej strony salonu



### 1 Pobierz pliki źródłowe

Pobierz pliki źródłowe przykładów z witryny <ftp://ftp.helion.pl/przyklady/hfhtcs.zip>. Kiedy już je pobierzesz i rozpakujesz, zajrzyj do katalogu `rozdzial2\salon`; znajdziesz w nim pliki `salon.html`, `drinki.html` oraz `jakdojehac.html` (oraz kilka plików z obrazami).



Strona salonu Head First powoli się rozrasta. Czy uważasz, że przechowywanie wszystkich plików w jednym katalogu jest dobrym pomysłem na zapewnienie właściwej organizacji witryny? Co byś zmodyfikował?

## 2 Wprowadź modyfikacje w pliku salon.html

Otwórz plik *salon.html* w swoim ulubionym edytorze. Dodaj do niego tekst oraz kod HTML, który na poniższym przykładzie został wyróżniony szarym tłem. Dalej — wpisz wyróżnione fragmenty; na następnej stronie sprawdzimy, jak to wszystko działa.

```
<html>
  <head>
    <title>Salon Head First</title>
  </head>
  <body>
    <h1>Witamy w nowym i poprawionym salonie Head First</h1>
    
    <p>
      Zaglądaj do nas każdego wieczoru,
      by rozkoszować się
      orzeźwiającymi <a href="drinki.html">drinkami</a>,
      miłą rozmową lub jedną bądź dwoma
      partyjkami gry <em>Dance Dance Revolution</em>.
      Zawsze zapewniamy
      bezprzewodowy dostęp do Internetu - ale pamiętaj!
      obowiązuje zasada
      PSWSW (przynieś swój własny serwer WWW).
    </p>
    <h2>Jak do nas trafić</h2>
    <p>
      Znajdziesz nas w samym centrum Webowic. Zapraszamy!
      Jeśli potrzebujesz pomocy, aby nas znaleźć, to zajrzyj na
      stronę ze szczegółowym <a href="jakdojehac.html">opisem dojazdu</a>.
    </p>
  </body>
</html>
```

Do nagłówka dodajmy tekst „nowym i poprawionym”.

W tych miejscach dodaliśmy kod HTML tworzący łącze ze stroną o naszych napojach.

Do tworzenia łącza używany jest element <a>; już zaraz szczegółowo opiszemy, jak on działa...

Tu musimy dodać fragment tekstu, który skieruje czytelnika na stronę z opisem dojazdu.

A tu umieściliśmy łącze do strony z opisem dojazdu; także w tym przypadku użyliśmy elementu <a>.

## 3 Zapisz plik salon.html i sprawdź, jak działa.

Kiedy skończysz wprowadzanie modyfikacji zapisz plik *salon.html* i otwórz go w przeglądarce. Poniżej wymieniliśmy, co powinieneś sprawdzić...

- ❶ Kliknij łącze „drinkami” — powinna się pojawić strona poświęcona naszym drinkom.
- ❷ Kliknij przycisk „Wstecz” w oknie przeglądarki, a ponownie zostanie wyświetlona główna strona salonu Head First.
- ❸ Kliknij łącze „opisem dojazdu” — powinna zostać wyświetlona strona z informacjami, jak nas znaleźć.

No dobrze.  
Zajrzałem na nową  
stronę salonu Head First,  
sprawdziłem łącza i wszystko  
świetnie działa. Niemniej  
jednak chciałbym mieć  
pewność, że rozumiem, jak  
działa ten nowy kod HTML.



Za kulisami

## Co zrobiliśmy?

- 1 Przeanalizujemy dokładnie proces tworzenia łącza w języku HTML. W pierwszej kolejności tekst, który ma być łączem, należy umieścić w elemencie `<a>` w następujący sposób:

`<a>drinkami</a>`



Do tworzenia łącza do innej strony  
potrzebny jest element `<a>`.

`<a>opisem dojazdu</a>`



Zawartość elementu `<a>` służy jako etykieta łącza.  
W przeglądarce etykieta ta jest wyróżniana podkreśleniem,  
co informuje użytkownika o tym, iż można ją kliknąć.

- 2 Po utworzeniu etykiet dla obu łączy musimy dodać do strony kod HTML, który poinformuje przeglądarkę o ich stronach docelowych:

`<a href="drinki.html">drinkami</a>`



To właśnie przy użyciu  
attributu `href` określana jest  
strona docelowa.

W przypadku tego łącza przeglądarka  
wyświetli etykietę „drinkami”, której  
kliknięcie spowoduje wyświetlenie  
strony `drinki.html`.

`<a href="jakdojehac.html">opisem dojazdu</a>`



Z kolei w przypadku tego  
łącza przeglądarka wyświetli  
etykietę „opisem dojazdu”,  
której kliknięcie spowoduje  
wyświetlenie strony  
`jakdojehac.html`.

## Co robi przeglądarka?

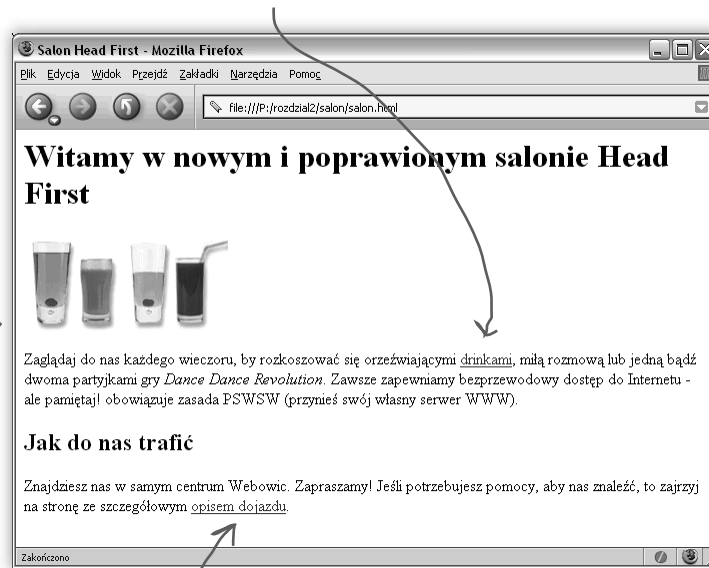
- ① Przede wszystkim jeśli podczas wyświetlania strony przeglądarka napotka element `<a>`, wyświetli jego zawartość jako etykietę, którą użytkownik będzie mógł kliknąć.



Za kulisami

```
<a href="drinki.html">drinkami</a>
```

Słowa „drinkami” i „opisem dojazdu” są umieszczone pomiędzy znacznikami elementu `<a>`, a zatem zostaną przez przeglądarkę wyświetlone jako etykiety, które użytkownik może kliknąć.



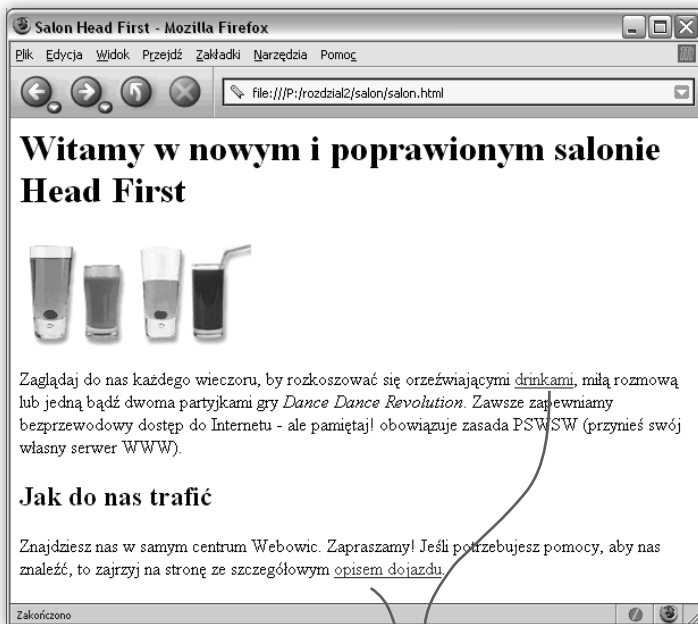
```
<a href="jakdojehac.html">opisem dojazdu</a>
```

**Użyj elementu `<a>`, aby utworzyć hipertekstowe łącze do innej strony WWW. Zawartość elementu zostanie wyświetlona na stronie, a użytkownik będzie mógł ją kliknąć i przejść do strony docelowej. Strona docelowa łącząca jest określana przy użyciu atrybutu href.**



## Jak działają łącza

- ② Później, kiedy użytkownik kliknie łącze, przeglądarka określi jego stronę docelową na podstawie wartości atrybutu „href”.



Użytkownik klika łącze „drinkami” lub...

... łącze „opisem dojazdu”.

Kiedy użytkownik kliknie łącze „opisem dojazdu”, przeglądarka pobiera wartość jego atrybutu href, którą w tym przypadku jest jakdojechac.html...

`<a href="drinki.html">drinkami</a>`

Jeśli użytkownik kliknął łącze „drinkami”, to przeglądarka pobierze wartość jego atrybutu href, którą jest drinki.html...

... i tądże stronę drinki.html.

`<a href="jakdojechac.html">opisem dojazdu</a>`

... i tądże stronę jakdojechac.html.



## Atrybuty

Atrybuty pozwalają na podawanie dodatkowych informacji o elementach. Choć jeszcze nie opisaliśmy ich dokładnie, to jednak miałeś już okazję poznać kilka z nich:



`<style type="text/css">` ← Atrybut `type` określa używany język arkuszy stylów, w tym przypadku jest to CSS.

`<a href="irule.html">` ← Atrybut `href` wskazuje stronę docelową łącza.

`` ← Atrybut `src` określa nazwę pliku, który ma zostać wyświetlony przez znacznik `img`.

Przeanalizujemy hipotetyczny przykład, który jeszcze lepiej uzmysłowi Ci, jak działają atrybuty:

### A co by było, gdyby istniał element `<samochód>`:

Otóż, gdyby istniał element `<samochód>`, to na pewno chciałbyś napisać następujący fragment kodu:

`<samochód>Mój mały mini</samochód>` ← W przypadku braku atrybutów jedyną informacją, jaką możemy podać, jest opisowa nazwa samochodu.

Jednak znacznik `<samochód>` w takiej postaci daje nam jedynie możliwość określenia opisowej nazwy samochodu — nie zawiera natomiast żadnych informacji na temat marki, modelu, także tego, czy auto jest kabrioletem, ani setek innych szczegółowych informacji, które mogą nas interesować. Gdyby więc faktycznie istniał element `<samochód>`, to moglibyśmy zastosować następujące przykładowe atrybuty:

Jednak dzięki zastosowaniu atrybutów możemy dostosować element i umieścić w nim wiele różnych informacji.

`<samochód marka="BMW" model="Mini Cooper" kabriolet="Nie">Mój mały mini</samochód>`

To wygląda znacznie lepiej, nieprawdaż? Powyższy kod przekazuje nam znacznie więcej informacji zapisanych w wygodnej i przejrzystej postaci.



### BEZPIECZEŃSTWO PRZEDE WSZYSTKIM

Atrybuty są zawsze zapisywane w taki sam sposób: w pierwszej kolejności podaje się nazwę atrybutu, następnie znak równości, a następnie wartość atrybutu zapisaną pomiędzy znakami cudzysłowu.

Czasami można się spotkać z dokumentami HTML, w których cudzysłowy wokół wartości atrybutów są pomijane, jednak warto zdobyć się na tę odrobinę wysiłku i je zapisywać. Okazuje się bowiem, że takie niemądre lenistwo może Ci potem przysporzyć wielu problemów (o czym przekonasz się w dalszej części książki).

### Pisz tak (postać poprawna)

`<a href="top10.html">Najlepsze filmy</a>`

Diagram explaining the structure of the correct code: `<` is labeled 'znak równości', `a` is 'nazwa atrybutu', `href="` is 'cudzysłów', `top10.html` is 'wartość atrybutu', and `>` is 'cudzysłów'.

### A nie tak (postać błędna)

`<a href=top10.html>Najlepsze filmy</a>`

**BŁĄD** — wartość atrybutu nie została zapisana pomiędzy znakami cudzysłowu.

### Nie ma niemądrych pytań

#### **❓ Czy mogę sam tworzyć nowe atrybuty dla istniejących elementów HTML?**

**U.** Nie, gdyż przeglądarki WWW rozpoznają jedynie predefiniowany zbiór atrybutów dla poszczególnych elementów HTML. Gdybyś sam wymyślił jakiś atrybut, to przeglądarka i tak nie wiedziałaby, co należy z nim zrobić, co więcej, jak się wkrótce przekonasz, takie postępowanie bardzo łatwo może wpędzić Cię w tarapaty. Kiedy przeglądarka jest w stanie rozpoznać element lub atrybut, to mówimy, iż jest on „obsługiwany” bądź też, że przeglądarka go „obsługuje”. Powinieneś używać wyłącznie obsługiwanych elementów i atrybutów.

#### **❓ A kto decyduje, jakie atrybuty są „obsługiwane”?**

**U.** Istnieją komisje zajmujące się standardami i to one określają elementy i atrybuty HTML. Komisje te składają się z osób, które nie mają nic lepszego do roboty które wspaniałomyślnie ofiarowują swój czas i wysiłek, by zadbać o istnienie jednej „mapy drogowej” języka HTML, implementowanej później we wszystkich przeglądarkach.

#### **❓ Skąd mogę dowiedzieć się, jakie elementy i atrybuty są obsługiwane? Ewentualnie, czy wszystkie atrybuty można stosować we wszystkich elementach?**

**U.** W każdym elemencie można używać jedynie określonego zbioru atrybutów. Wyobraź to sobie w następujący sposób: czy użyłbyś atrybutu „kabriolet” w elemencie <to s t e r>? Zapewne nie! A zatem będziesz chciał używać jedynie atrybutów, które mają sens w kontekście konkretnego elementu i które są obsługiwane. Podczas dalszej lektury niniejszej książki oraz omawiania kolejnych zagadnień będziesz się uczyć, jakie atrybuty są obsługiwane przez poszczególne elementy. Natomiast kiedy już zakończysz lekturę, będziesz mógł sobie odświeżyć pamięć, korzystając z wielu doskonałych publikacji, takich jak książka *HTML i XHTML. Przewodnik encyklopedyczny* (wydana przez Wydawnictwo Helion).





## Atrybuty bez tajemnic

W tym tygodniu:

Wyznania atrybutu href

**HeadFirst:** Witamy, href. Bez wątpienia to wielka przyjemność prowadzić wywiad z tak znaczącym atrybutem jak ty.

**href:** Dziękuję. Dobrze być tu i choć na chwilę oderwać się od tego całego łączenia. Ono naprawdę może doprowadzić atrybut na skraj wyczerpania. Zgadnij, kto mówi przeglądarce, co ma zrobić za każdym razem, gdy ktoś kliknie łącze. To właśnie ja.

**HeadFirst:** Jesteśmy wdzięczni, że znalazłeś dla nas trochę czasu w swoim niezwykle napiętym terminarzu. Ale dlaczego nie zaczniesz opowiadać od samego początku? Jak to jest być atrybutem?

**href:** No tak. Jasne. Otóż atrybuty służą do dostosowywania działania elementów. Nie ma nic trudnego w umieszczeniu fragmentu zawartości strony, na przykład „Zapisz się natychmiast!”, pomiędzy znacznikami elementu, choćby takiego jak `<a>`. Wystarczy zapisać coś takiego `<a>Zapisz się natychmiast!</a>`. Jednak beze mnie, atrybutu href, nie masz żadnej możliwości przekazania elementowi `<a>` informacji o stronie docelowej.

**HeadFirst:** Jak na razie wszystko jasne...

**href:** ...jednak dzięki atrybutom można określać dodatkowe informacje dotyczące elementów. W moim przypadku informacja ta dotyczy adresu strony docelowej: `<a href="rejestracja.html">Zapisz się natychmiast!</a>`. Ten kod oznacza, że element `<a>` o etykiecie „Zapisz się natychmiast!” jest połączony ze stroną *rejestracja.html*. Oczywiście istnieje bardzo wiele innych atrybutów, jednak ja jestem używany w elementach `<a>` w celu określenia strony docelowej łącza.

**HeadFirst:** Świetnie. Teraz muszę zadać to pytanie i mam tylko nadzieję, że nie poczujesz się urażony; ale co w zasadzie oznacza twoja nazwa? href? Co to w ogóle jest?

**href:** To jedna z najstarszych nazw w Internecie. Oznacza ona „odwołanie hipertekstowe” (ang. *hypertext reference*). Jednak znajomi używają skróconej nazwy — href.

**HeadFirst:** Czyli?

**href:** Odwołanie hipertekstowe to po prostu inne określenie zasobu znajdującego się na lokalnym komputerze lub gdzieś w Internecie. Zazwyczaj zasobem tym jest strona WWW, jednak nic nie stoi na przeszkodzie by był to klip dźwiękowy, wideo lub cokolwiek innego.

**HeadFirst:** To bardzo ciekawe. Nasi czytelnicy widzieli do tej pory jedynie łącza wskazujące na inne stworzone przez nich strony; w jaki sposób tworzy się łącza do innych stron lub plików znajdujących się na WWW?

**href:** Ech... chyba muszę już wracać do roboty — Sieć jest taka bezsensowna beze mnie. Poza tym uczenie czytelników HTML-a to chyba wasze zadanie, nieprawdaż?

**HeadFirst:** Już dobrze, dobrze. Tak, zajmiemy się tym zaraz. Dziękujemy, że nas odwiedziłeś, href.



### Ćwiczenia

Wcześniej utworzyłeś łącza ze strony *salon.html* do stron *drinki.html* oraz *jakdojechać.html*. Teraz stworzysz łącza prowadzące w odwrotnym kierunku. Poniżej przedstawiony został kod HTML strony *drinki.html*. Na dole tej strony dodaj łącza o etykiecie „Powrót na stronę salonu”, które będzie prowadzić na stronę *salon.html*.

```
<html>
  <head>
    <title>Firmowe napoje salonu Head First</title>
  </head>
  <body>
    <h1>Nasze napoje firmowe</h1>
    <h2>Zielona herbata orzeźwiająca</h2>
    <p>
      
      Szklanica pełna witamin i minerałów. Ten napój to źródło
      zdrowia, jakie zapewnia zielona herbata oraz mieszanka płatków
      rumianku i korzenia imbiru.
    </p>
    <h2>Koncentracja malinowej ochłody</h2>
    <p>
      
      Ten orzeźwiający napój, stanowiący połączenie soku malinowego
      z trawą cytrynową, skórką cytrynową oraz owocami głogu, sprawi,
      że Twój umysł stanie się bystry i chłonny.
    </p>
    <h2>Niebiańska borówka</h2>
    <p>
      
      Wstrząśnięta esencja jagód i wiśni, zmieszana z herbatą
      z czarnego bzu błyskawicznie, sprawi, że poczujesz błogą rozkosz.
    </p>
    <h2>Eksplzja żurawinowego antyutleniacza</h2>
    <p>
      
      Otrzeźwiej, wdychając aromaty żurawin i hibiskusa zawarte
      w lekkim napoju pełnym witaminy C.
    </p>
  </body>
</html>
```

Tutaj powinieneś  
umieścić nowy  
fragment kodu.



Kiedy skończysz, podobnie zmodyfikuj plik *jakdojechać.html*.



Potrzebujemy pomocy przy tworzeniu i analizie elementów `<a>`. Zważywszy na Twoją nową wiedzę dotyczącą tego elementu, mamy nadzieję, że będziesz nam w stanie pomóc. W każdym wierszu przedstawionej poniżej tabeli znajdziesz pewną kombinację etykiety, strony docelowej oraz elementu `<a>`. Uzupełnij je o wszelkie brakujące informacje. Pierwszy wiersz uzupełniliśmy sami.

Etykieta	Strona docelowa	Kod HTML
Gorący czy nie?	<code>goracy.html</code>	<code>&lt;a href="goracy.html"&gt;Gorący czy nie?&lt;/a&gt;</code>
Życiorys	<code>cv.html</code>	
	<code>cukiereczek.html</code>	<code>&lt;a href="_____ "&gt;Cukiereczek&lt;/a&gt;</code>
Oto ja w moim mini	<code>mini-cooper.html</code>	
Zagrajmy		<code>&lt;a href="szczesciarz.html"&gt; _____ &lt;/a&gt;</code>

### Nie ma niemądrych pytań

**P.** Widziałem wiele stron, na których łącza miały postać graficzną, a nie tekstową. Czy także w takich przypadkach można zastosować element `<a>`?

**O.** Owszem, jeśli umieścisz element `<img>` pomiędzy znacznikami elementu `<a>`, to obraz stanie się łączem i będzie go można kliknąć, analogicznie jak dzieje się to w przypadku tekstu. O obrazkach będziemy pisali dopiero w dalszej części książki, jednak powinieneś wiedzieć, że z powodzeniem można ich używać jako łącza.

**P.** Czy to oznacza, że wewnątrz elementu `<a>` można umieszczać dowolną zawartość, która w ten sposób stanie się łączem? Czy w elemencie `<a>` można umieścić na przykład akapit tekstu?

**O.** Poczekaj. Nie tak szybko. Wewnątrz elementu `<a>` nie można umieszczać dowolnych innych elementów. Ogólnie rzecz biorąc, umieszcza się w nim wyłącznie tekst oraz obrazy (jak również dowolne ich kombinacje). Problem możliwości wzajemnego zagnieżdżenia znaczników, czyli umieszczania jednych elementów wewnątrz innych, jest całkowicie odrębnym zagadnieniem. Ale nie przejmuj się, już wkrótce się nim zajmiemy.

Twoja praca nad witryną salonu Head First naprawdę się opłaciła. Dzięki tym kuszącym napojom i dokładnemu opisowi dojazdu do lokalu wiele osób zaczęło regularnie nas odwiedzać i zaglądać na naszą witrynę WWW. Teraz pojawiły się plany znacznego rozszerzenia zawartości naszej witryny.



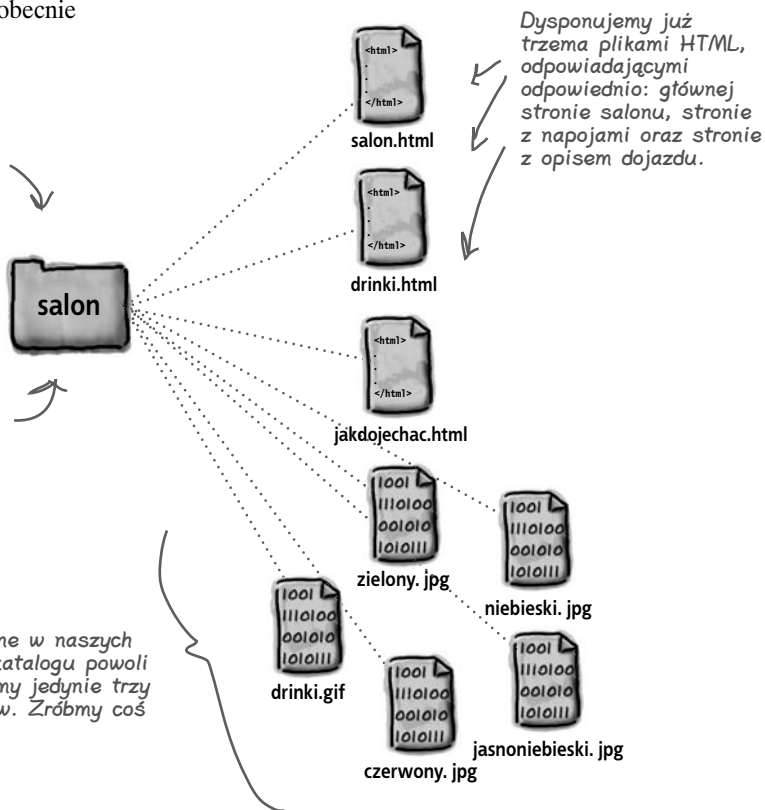
## Organizowanie zawartości witryny

Zanim zaczniesz tworzyć kolejne strony WWW, zadбай o ich odpowiednie zorganizowanie. Do tej pory umieszczaliśmy wszystkie nasze pliki, zarówno strony, jak i obrazy, w jednym katalogu. Bardzo szybko sam dojdiesz do wniosku, iż nawet w przypadku witryny WWW średniej wielkości, rozmieszczenie plików HTML, obrazów oraz wszelkich innych zasobów w kilku odrębnych katalogach znacznie ułatwia zarządzanie nimi. Oto, jak obecnie wygląda zawartość naszej witryny.

Stworzyliśmy główny katalog o nazwie salon, w którym umieszczaliśmy wszystkie pliki wchodzące w skład witryny.

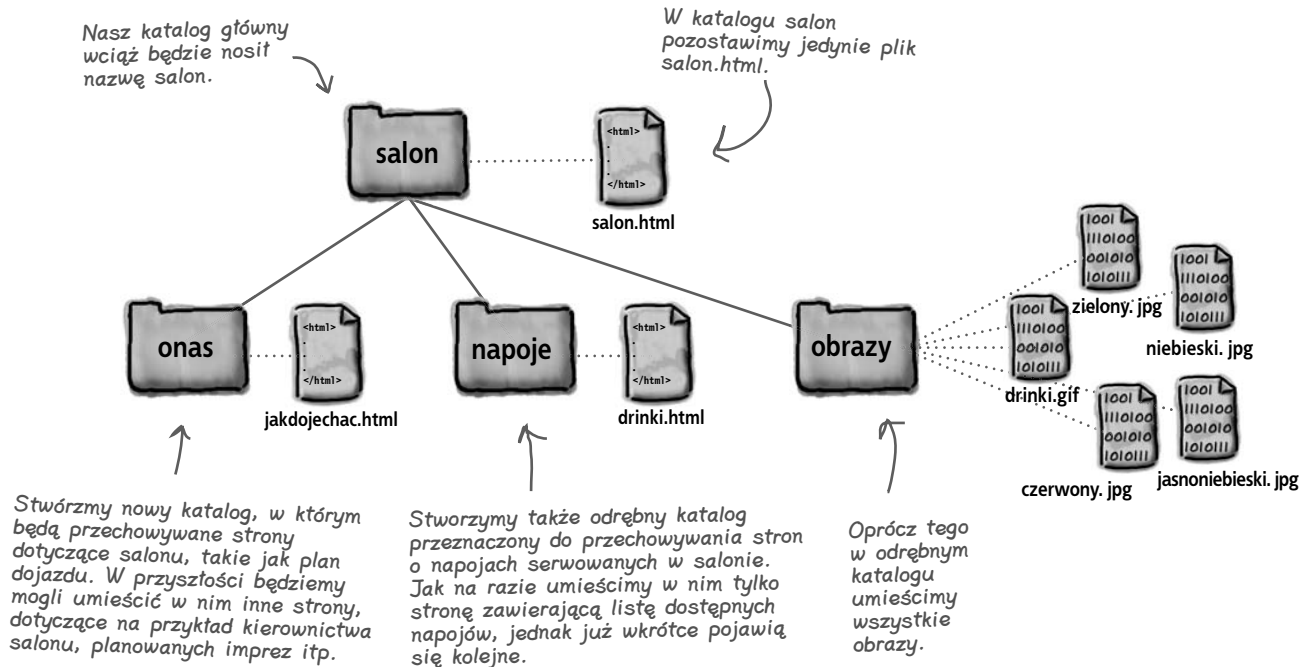
Bardzo często taki katalog jest określany jak „główny katalog witryny”, co oznacza, że jest to najwyższy katalog w hierarchii, zawierający wszystkie zasoby wchodzące w skład witryny.

A to są wszystkie obrazy używane w naszych stronach. Jak widać, w naszym katalogu powoli zaczyna się robić bałagan, a mamy jedynie trzy dokumenty HTML i kilka obrazków. Zróbmy coś z tym problemem.



## Organizacja salonu...

Spróbujemy zatem stworzyć jakiś sensowny sposób organizacji salonu Head First. Powinieneś przy tym pamiętać, że każdą witrynę można zorganizować na wiele różnych sposobów. Zaczniemy od prostego rozwiązania i stworzymy kilka katalogów dla stron WWW. Oprócz tego wszystkie obrazy umieścimy w jednym miejscu.



### Nie ma niemądrych pytań

**P.** Skoro tworzyliśmy jeden katalog przeznaczony do przechowywania obrazów, to dlaczego nie można by stworzyć katalogu o nazwie „html”, w którym byłyby umieszczane dokumenty HTML?

**O.** Oczywiście można by. Nie ma żadnych „jedynie słusznych” sposobów określania organizacji witryn WWW. Zazwyczaj będziesz się starał stosować taki sposób organizacji, który najbardziej odpowiada Tobie oraz Twoim użytkownikom. Podobnie jak w przypadku wszelkich innych decyzji projektowych będziesz się zapewne starał zastosować schemat organizacji, który jest prosty, a jednocześnie na tyle elastyczny, by pozwolił na rozbudowę witryny.

**P.** A może by tak umieścić katalog służący do przechowywania obrazów we wszystkich pozostałych katalogach, takich jak *napoje* lub *onas*?

**O.** Także w tym przypadku można by tak zrobić. Można się jednak spodziewać, że niektóre spośród obrazów będą używane na kilku stronach i dlatego umieściliśmy je wszystkie w jednym katalogu. Jeśli jednak tworzysz witrynę, w której do poszczególnych części przypisanych jest wiele obrazów, to możesz zdecydować się na stworzenie katalogu na obrazy w każdej z „gałęzi”.

**P.** „W każdej z gałęzi”?

**O.** Sposób organizacji katalogów można najprościej zrozumieć, wyobrażając je sobie jako drzewo odwrócone korzeniem do góry. Na samej górze znajduje się katalog główny, a każda ścieżka prowadząca w dół, do pliku lub katalogu, stanowi „gałąź”.







### Ćwiczenia

Twoim aktualnym zadaniem jest stworzenie struktury plików i katalogów przedstawionej na poprzedniej stronie. Poniżej dokładnie opisaliśmy, co musisz zrobić:

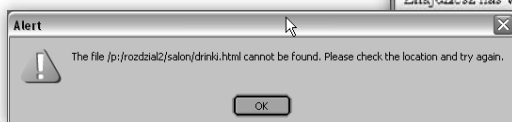
- 1 Odszukaj katalog *salon* i utwórz w nim nowe podkatalogi o nazwach *onas*, *napoje* oraz *obrazy*.
- 2 Przenieś plik *jakdojechać.html* do katalogu *onas*.
- 3 Przenieś plik *drinki.html* do katalogu *napoje*.
- 4 Przenieś wszystkie pliki obrazów do katalogu *obrazy*.
- 5 W końcu wyświetl w przeglądarce stronę *salon.html* i sprawdź, czy działają umieszczone na niej łącza. Porównaj bieżące wyniki z tymi uzyskanymi wcześniej.

## Problemy techniczne

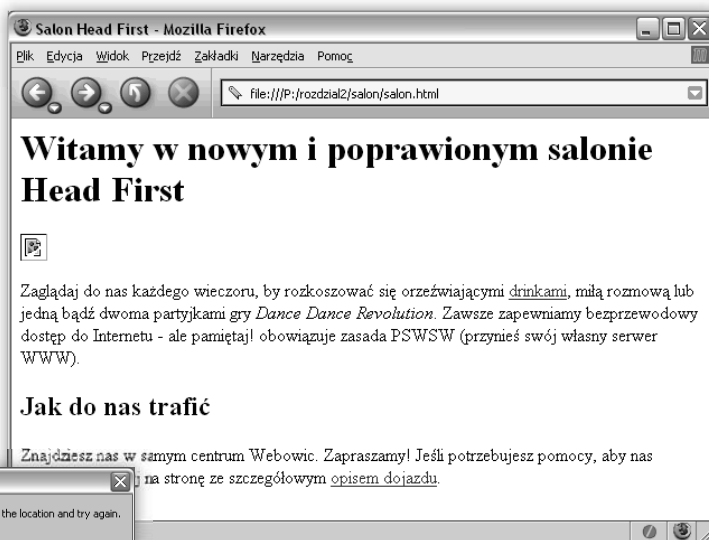
**Wygląda na to, że po przeniesieniu stron do podkatalogów pojawiły się nowe problemy techniczne.**

Jak się okazuje, obecnie na stronie nie wyświetla się jeden z umieszczonych na niej obrazów. Jest to przykład tak zwanego „zerwanego łącza”.

Co więcej, okazuje się, że znacznie gorsze rzeczy dzieją się, gdy klikniesz łącze do strony napojów firmowych (lub opisu dojazdu) — na ekranie pokazuje się okno dialogowe informujące, że nie można znaleźć strony.



Niektóre przeglądarki wyświetlają ten komunikat w postaci strony WWW, a nie okna dialogowego.



Jak sądzę problem polega na tym, iż przeglądarka uważa, że wszystkie pliki wciąż znajdują się w tym samym katalogu co strona *salon.html*. Musimy zatem zmienić łańcuch, by wskazywały na pliki przeniesione do nowych katalogów.



### **Bardzo słusznie. Musimy poinformować przeglądarkę o nowym położeniu plików.**

Do tej pory wartości podane w atrybutach `href` wskazywały na strony znajdujące się w *tych samych* katalogu. Jednak zazwyczaj witryny są nieco bardziej złożone, więc musi istnieć możliwość odwoływania się do stron umieszczonych w *innych katalogach*.

W tym celu należy określić ścieżkę prowadzącą z danej strony do strony docelowej. Może to oznaczać konieczność przejścia jeden lub dwa poziomy niżej w hierarchii katalogów bądź wyjścia na któryś z wyższych poziomów. W obu tych przypadkach możemy się posłużyć *ścieżką względną*, którą podamy jako wartość atrybutu `href`.

### Planowanie ścieżek...

Zastanów się, co robisz, gdy planujesz wakacje w rodzinnym wozie kempingowym? Bierzesz mapę, znajdujesz swoje bieżące położenie, a następnie zaznaczasz trasę do miejsca docelowego. Opis trasy oraz instrukcje, jak należy jechać, są zatem określane *względem* bieżącego położenia. Czyli gdybyś przebywał w innym mieście, instrukcje te wyglądałyby zupełnie inaczej, nieprawdaż?



No dobra, tak naprawdę to pewnie zająłbyś do serwisu Google Maps, ale tym razem możesz nam chyba pójść na rękę.

Określając względną ścieżkę na potrzeby tworzonych łączy, powinieneś postępować dokładnie w taki sam sposób: zacząć od strony, na jakiej chcesz umieścić łącze, a następnie przejść przez wszystkie katalogi znajdujące się na drodze do strony docelowej.

Istnieją także inne rodzaje ścieżek. Przedstawimy je w następnych rozdziałach.

Przeanalizujemy teraz kilka ścieżek względnych (a jednocześnie poprawimy łącza umieszczone na stronie salonu Head First):

### Łącza do zasobów znajdujących się w katalogach podrzędnych

- 1 Określenie łącza ze strony *salon.html* do *drinki.html*.

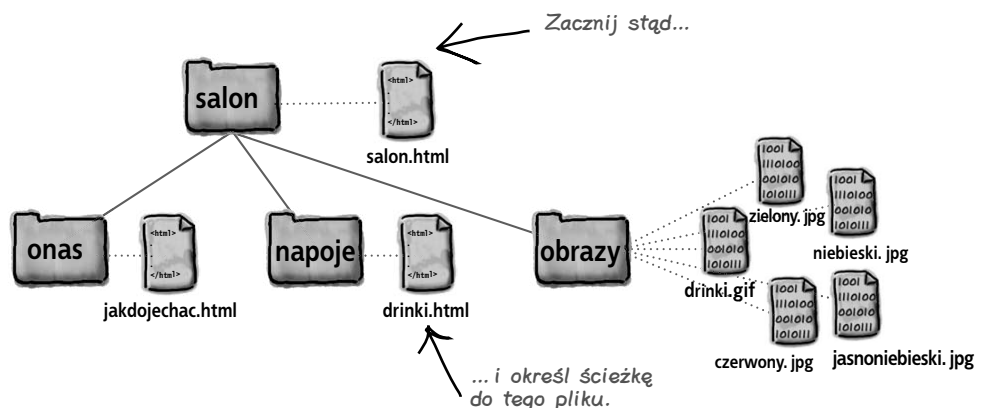
Musimy poprawić łącza prowadzące ze strony *salon.html* na stronę *drinki.html*. Oto, jak aktualnie wygląda element `<a>` umieszczony w kodzie strony:

```
<a href="drinki.html">drinkami</a>
```

Aktualnie używamy jedynie nazwy pliku *drinki.html*, co informuje przeglądarkę, iż strony docelowej należy szukać w tym samym katalogu, w którym znajduje się bieżąca strona *salon.html*.

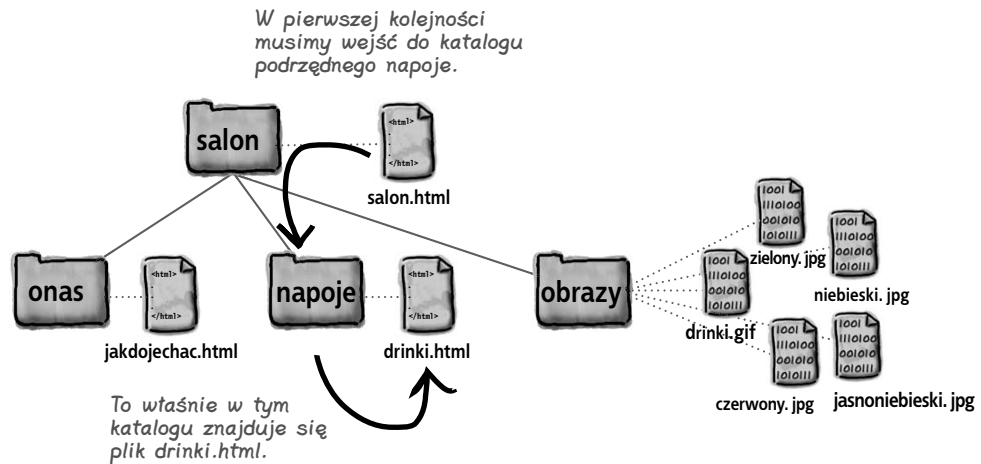
- 2 Określenie strony początkowej oraz docelowej.

Podczas reorganizacji stron salonu Head First stronę *salon.html* pozostawiliśmy w katalogu głównym — *salon* — natomiast stronę *drinki.html* przenieśliśmy do podkatalogu o nazwie *napoje*.



### 3 Wyznacz ścieżkę ze strony początkowej do strony docelowej.

A teraz wyznaczmy ścieżkę. Aby dotrzeć ze strony *salon.html* do strony *drinki.html*, musimy w pierwszej kolejności wejść do katalogu *napoje*, a następnie odszukać w nim plik *drinki.html*.



### 4 Utwórz atrybut href, w którym podasz wyznaczoną ścieżkę.

Kiedy już określiliśmy ścieżkę, musimy ją zapisać w postaci zrozumiałej dla przeglądarki. Sposób zapisu tej ścieżki przedstawia się następująco:

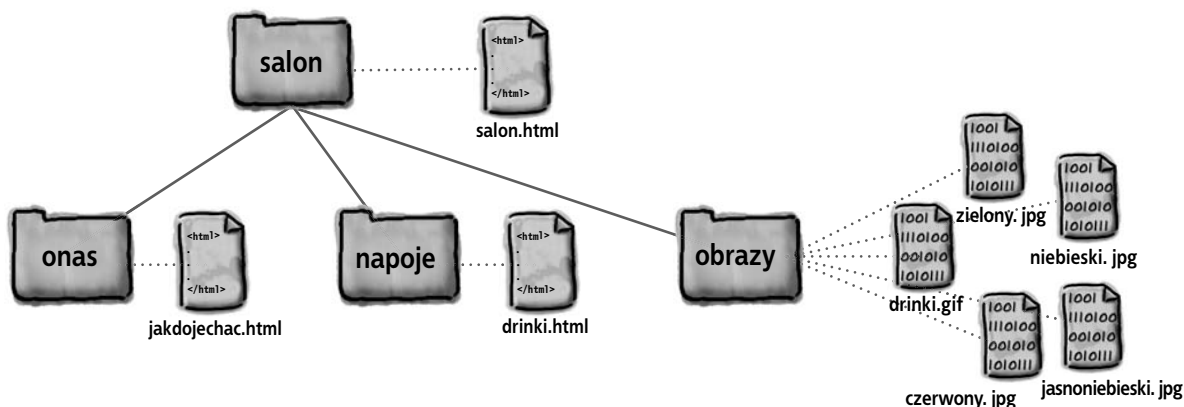


Wyznaczoną ścieżkę względną zapisaliśmy jako wartość atrybutu href. Gdy klikniemy teraz łącze, przeglądarka spróbuje znaleźć plik *drinki.html*, który powinien się znajdować w katalogu *napoje*.

### Zaostrz ołówek



Teraz Twoja kolej; określ względną ścieżkę od pliku *salon.html* do pliku *jakdojechac.html*. Kiedy już to zrobisz, uzupełnij podany poniżej element `<a>`. Sprawdź poprawność swoich odpowiedzi, porównując je z odpowiedziami podanymi na końcu rozdziału, a następnie zmodyfikuj oba elementy `<a>` umieszczone w pliku *salon.html*.



`<a href="_____ ">opisem dojazdu</a>`

TU WPISZ SWOJĄ ODPOWIEDŹ ↗

## Podróż w przeciwnym kierunku — łącze do katalogu nadrzędnego

- 1 Tworzenie łącza ze strony *jakdojechac.html* do strony *salon.html*.

Pozostaje nam poprawić łącza „Powrót na stronę salonu”.

Oto aktualna postać elementu `<a>` znajdującego się w kodzie strony *jakdojechac.html*:

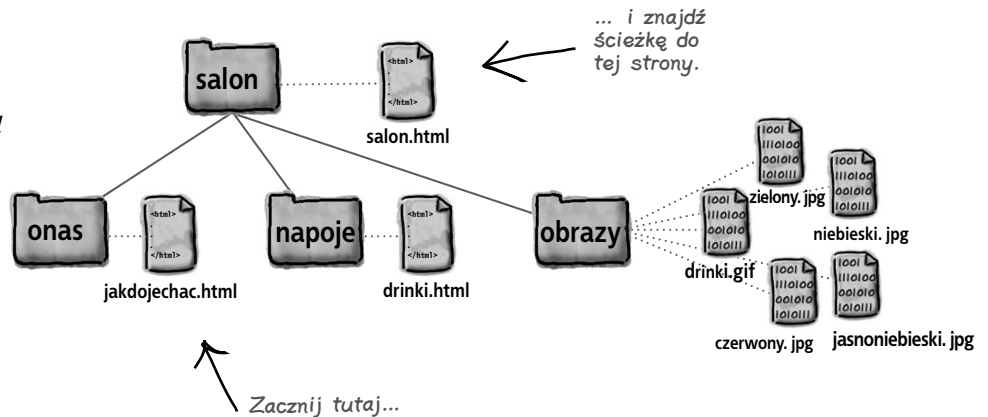
`<a href="salon.html">Powrót na stronę salonu</a>`

Aktualnie używamy jedynie nazwy strony *salon.html*, co sprawia, że przeglądarka poszukuje strony docelowej w tym samym katalogu, w którym znajduje się plik *jakdojechac.html*. Takie łącze nie może zatem działać poprawnie.

- 2 Określ stronę początkową oraz docelową.

Przyjrzyj się położeniu strony początkowej i docelowej.

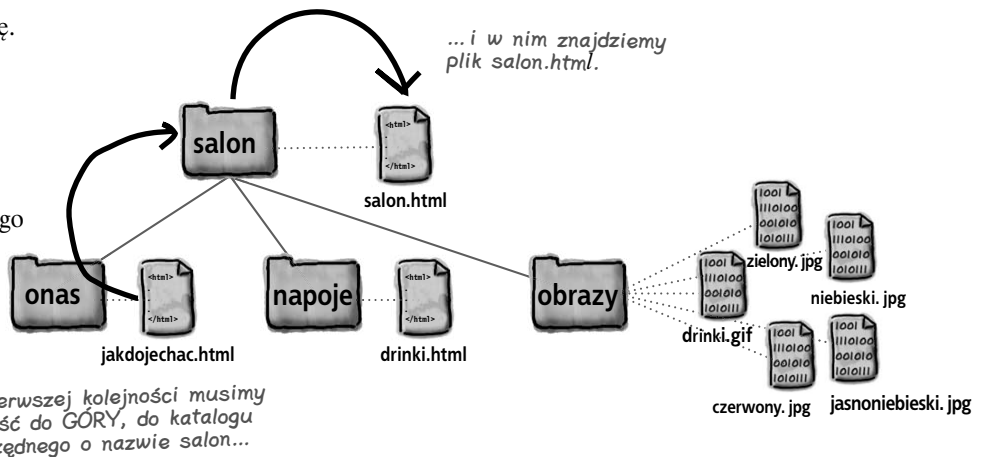
W tym przypadku stroną początkową jest plik *jakdojechac.html* umieszczony w katalogu *onas*. Z kolei stroną docelową jest plik *salon.html* umieszczony w katalogu *salon*, czyli nadrzędnym względem katalogu, w jakim jest przechowywany plik *jakdojechac.html*.



- 3 Wyznacz ścieżkę od strony początkowej do docelowej.

A teraz wyznacz ścieżkę.

Aby dostać się do strony *salon.html*, zaczynając od strony *jakdojechac.html*, musimy przejść do katalogu nadrzędnego i w nim znajdziemy nasz plik docelowy.



## Tworzenie wartości atrybutu href

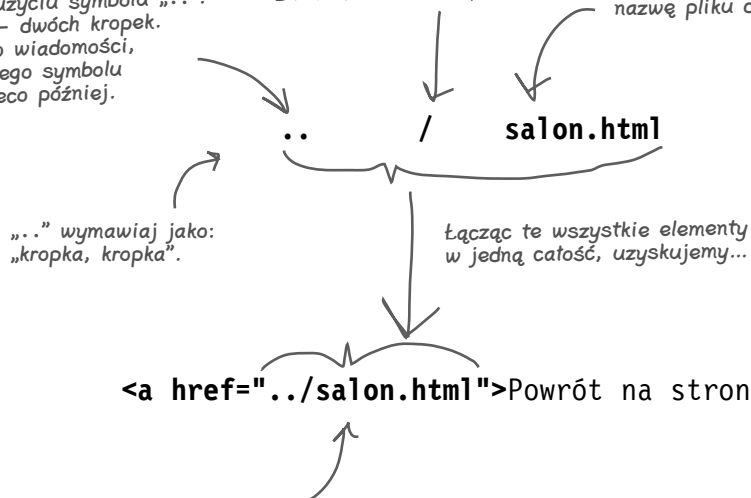
- 4 Podaj wartości atrybutu href reprezentującą wyznaczoną ścieżkę.

To już prawie wszystko. Po określeniu ścieżki pozostało Ci jedynie zapisać ją w postaci zrozumiałej dla przeglądarki. Zajmijmy się tym:

W pierwszej kolejności musisz przejść o jeden poziom do góry — do katalogu nadrzędnego. Jak to zrobić? Przy użyciu symbolu „..”. Tak właśnie — dwóch kropek. Przyjmij to do wiadomości, a znaczenie tego symbolu wyjaśnimy nieco później.

Poszczególne części ścieżki oddziel do siebie znakami ukośnika (/).

Na samym końcu podaj nazwę pliku docelowego.



Kiedy klikniesz to łącze, przeglądarka poszuka w katalogu nadrzędnym pliku *salon.html*.

Kropka, kropka

~~Do góry~~, w dół, artykuły gospodarstwa domowego, bielizna damska?



## Nie ma niemądrych pytań

**P.** Co to jest katalog nadrzędny? Jeśli mam katalog „owoce”, a w nim katalog „jabłka”, to czy katalog „owoce” jest katalogiem nadrzędnym katalogu „jabłka”?

**O.** Dokładnie. Katalogi (które czasami są także nazywane folderami) są bardzo często opisywane na podstawie ich wzajemnego położenia. Posługując się przykładem z pytania, można powiedzieć, że katalog „owoce” jest katalogiem nadrzędnym katalogu „jabłka”, natomiast katalog „jabłka” jest katalogiem podrzędnym katalogu „owoce”. Gdyby istniał jeszcze jeden katalog, na przykład „gruszki”, umieszczony w katalogu „owoce”, to byłby on katalogiem „sąsiednim” względem katalogu „jabłka”.

**P.** No dobrze, katalogi nadrzędne i podrzędne mają jeszcze jakiś sens, ale te dwie kropki?

**O.** Kiedy musisz poinformować przeglądarkę, że poszukiwany plik znajduje się w katalogu nadrzędnym, używasz właśnie symbolu `..`, który oznacza: „Przejdź DO GÓRY, do katalogu nadrzędnego”. Innymi słowy, te dwie kropki to w języku przeglądarki określenie katalogu nadrzędnego.

W naszym przykładzie chcieliśmy utworzyć łącze prowadzące ze strony *jakdojechac.html* znajdującej się w katalogu *onas* i wskazujące na stronę *salon.html* znajdującą się w katalogu *salon* — nadrzędnym względem katalogu *onas*. A zatem musieliśmy poinformować przeglądarkę, by szukała strony docelowej w katalogu, który w hierarchii jest położony o jeden poziom WYŻEJ. Dwie kropki są symbolem nakazującym przeglądarce przejście właśnie o jeden poziom WYŻEJ.

**P.** A co należy zrobić, gdy musimy przejść nie o jeden, lecz o dwa poziomy wyżej?

**O.** Dla każdego katalogu nadrzędnego do jakiego chcesz przejść, możesz użyć symbolu `...`. Każde użycie tego symbolu oznacza przejście o jeden poziom ku górze w hierarchii katalogów. Jeśli wobec tego chcesz przejść o dwa poziomy do góry, musisz użyć ścieżki o postaci: `../..`. Poszczególne elementy ścieżki zawsze należy oddzielać znakiem ukośnika (`/`); nie zapominaj o tym (bo przeglądarka nie wie, co oznacza symbol składający się z czterech kropek)!

**P.** Kiedy już znajdę się w katalogu o dwa poziomy wyżej, to jak mam poinformować przeglądarkę, gdzie ma szukać pliku docelowego?

**O.** Wystarczy połączyć ścieżkę `../..` z nazwą pliku. Jeśli zatem szukasz pliku „owoc.html” położonego dwa katalogi wyżej, cała ścieżka musi mieć postać: `../..owoc.html`. Być może przypuszczasz, że katalog określaną ścieżką `../..` nazywamy „dwukrotnie” nadrzędnym, jednak tak nie jest; używa się raczej określenia „katalog nadrzędny katalogu nadrzędnego” lub po prostu: `../..`.

**P.** Czy jest jakieś ograniczenie co do tego, jak wysoko można przejść, używając symbolu `..`?

**O.** Można w ten sposób przejść co najwyżej do poziomu katalogu głównego witryny WWW. W naszym przypadku katalogiem głównym jest *salon*. A zatem można przejść co najwyżej do niego.

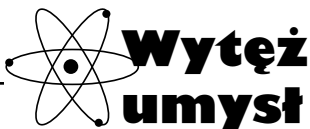
**P.** A co z poruszaniem się w przeciwnym kierunku — czy istnieje jakieś ograniczenie limitujące, ile katalogów w dół można zejść?

**O.** Cóż, to zależy tylko od ilości katalogów, jaka została utworzona. Jeśli stworzyłeś strukturę katalogów składającą się z 10 poziomów, to nic nie stoi na przeszkodzie, byś użył ścieżki schodzącej do katalogu znajdującego się na ostatnim — dziesiątym poziomie. Jednak nie zalecamy stosowania takich rozwiązań — pojawienie się tak ogromnej ilości poziomów w hierarchii katalogów będzie zapewne oznaczać, że przyjęty sposób organizacji witryny będzie zbyt skomplikowany.

Co więcej, istnieje ograniczenie ilości znaków, z jakich może się składać ścieżka — jej maksymalna długość wynosi 255 znaków. To bardzo dużo, więc jest mało prawdopodobne, byś kiedykolwiek potrzebował aż tylu znaków do zapisania swojej ścieżki, jednak jeśli tworzysz duże witryny, to jest to czynnik, o jakim należy pamiętać.

**P.** W moim systemie operacyjnym separatorem ścieżek jest znak lewego ukośnika (`\`) — czy to nie jego powinienem zatem używać?

**O.** Nie, przy zapisywaniu ścieżek w stronach WWW zawsze używamy znaku ukośnika (`/`). Nie wolno używać znaku lewego ukośnika (`\`). Wiele systemów operacyjnych używa innych znaków separatora (na przykład w systemie Windows używany jest znak lewego ukośnika `\`, a nie ukośnika `/`), jeśli jednak chodzi o ścieżki tworzone na potrzeby Sieci, to został wybrany jeden separator i jest on wszędzie konsekwentnie używany. A zatem niezależnie od tego, czy używasz systemu MacOS, Windows, Linux czy też jakiegoś innego, zapisując ścieżki w łączach umieszczanych na stronach WWW, zawsze powinieneś używać znaków ukośnika (`/`).



Teraz Twoja kolej: określ względną ścieżkę pomiędzy stronami *drinki.html* oraz *salon.html*, jaką należy umieścić w łączu „Powrót na stronę salonu”. Czym ścieżka ta różni się od ścieżki umieszczonej w pliku *jakdojechac.html*?

Niczym się nie różni — obie ścieżki są identyczne



### Poprawianie błędnych adresów obrazów...

Już niemal udało Ci się poprawić główną stronę salonu Head First i ponownie doprowadzić ją do działania; ostatnią rzeczą, jaką musisz poprawić, są błędne obrazy, które się nie wyświetlają.

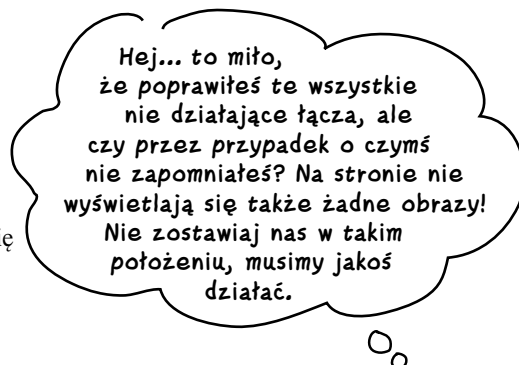
Nie przyglądaliśmy się jeszcze szczegółowo elementom `<img>` (zajmiemy się tym później), jednak jak na razie całkowicie wystarczy Ci informacja, że w atrybutach `src` tych elementów podawane są względne ścieżki — tak samo jak w atrybutach `href` elementów `<a>`.

Poniżej przedstawiliśmy postać elementu `<img>` podanego w kodzie strony `salon.html`:

```

```

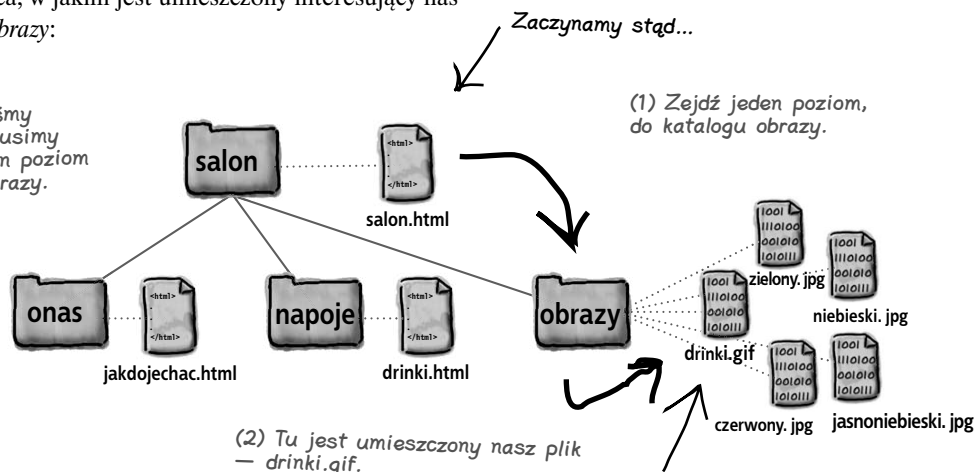
W tym miejscu podawana jest względna ścieżka określająca położenie obrazu. Ścieżkę tę określamy dokładnie tak samo jak w przypadku atrybutu `href` elementu `<a>`.



### Określanie ścieżki z pliku `salon.html` do obrazu `drinki.gif`

Aby określić tę ścieżkę, musimy przejść od strony początkowej — `salon.html` — do miejsca, w jakim jest umieszczony interesujący nas obraz, czyli do katalogu `obrazy`:

CEL: aktualnie jesteśmy w katalogu `salon` i musimy przesunąć się o jeden poziom niżej, do katalogu `obrazy`.



A zatem po połączeniu ze sobą elementów (1) i (2) z powyższego rysunku otrzymamy w efekcie ścieżkę o postaci „`obrazy/drinki.gif`”, zaś poprawna postać elementu `<img>` powinna być następująca:

```

```

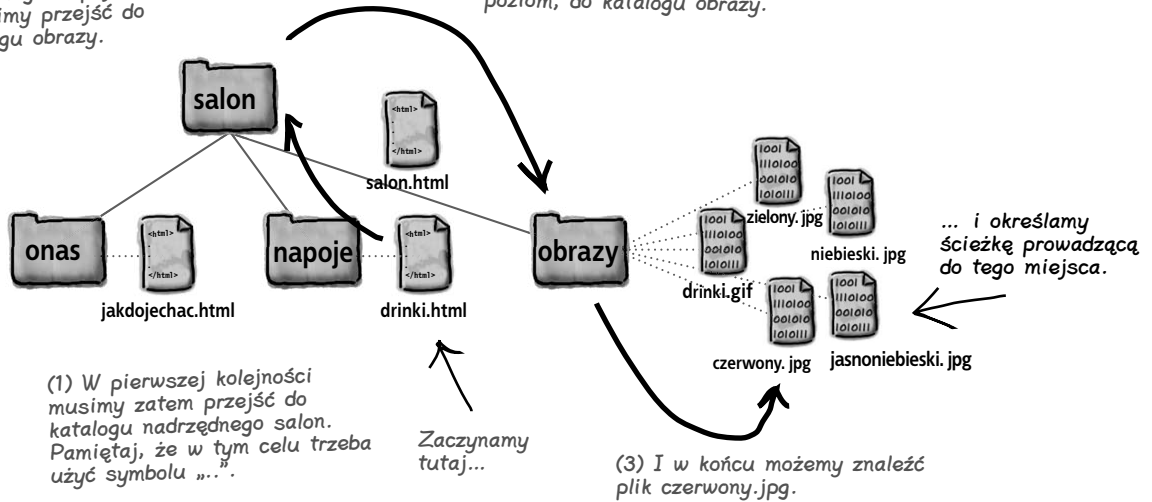
... i mamy określić ścieżkę do tego pliku.

## Określanie ścieżki od pliku *drinki.html* do obrazu *czerwony.jpg*

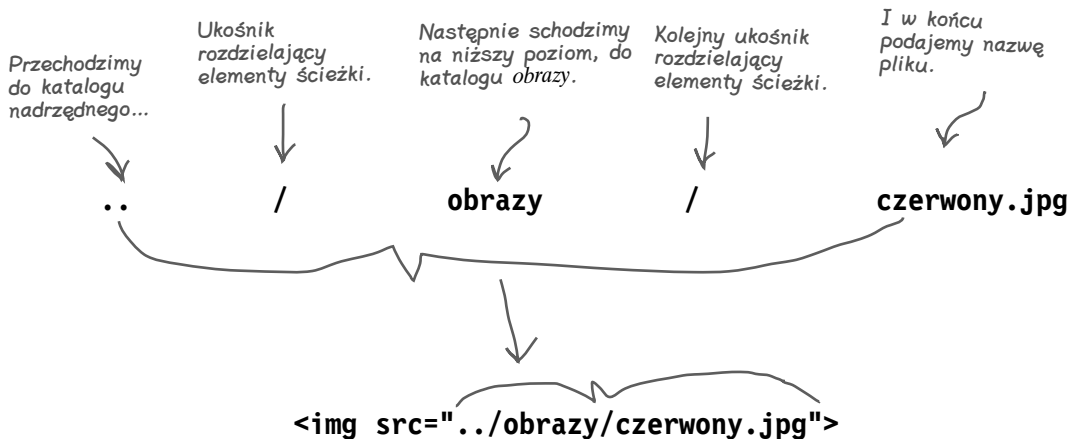
Na stronie z informacjami o napojach firmowych wyświetlanych jest kilka obrazów różnokolorowych drinków. Obrazy te są przechowywane w plikach: *czerwony.jpg*, *zielony.jpg* i tak dalej. Określmy najpierw ścieżkę do pliku *czerwony.jpg*; ścieżki do pozostałych obrazów będą bardzo podobne, gdyż wszystkie pliki graficzne zostały umieszczone w tym samym katalogu.

CEL: znajdujemy się w katalogu *napoje* i musimy przejść do katalogu *obrazy*.

(2) A następnie schodzimy jeden poziom, do katalogu *obrazy*.



Po połączeniu elementów (1), (2) oraz (3) uzyskujemy:

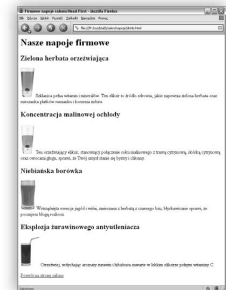




### Ćwiczenia

Wszystkie czynności, jakie wykonałeś do tej pory, pozwoliły poprawić łącza, które przestały działać w wyniku reorganizacji witryny salonu Head First. Niemniej jednak wciąż pozostają Ci jeszcze do poprawienia obrazy wyświetlane na stronach *salon.html* oraz *drinki.html*. Poniżej opisałeś, co musisz zrobić, aby je poprawić:

- 1 W pliku *salon.html* zmodyfikuj atrybut **src**, wpisując w nim ścieżkę `obrazy/drinki.gif`.
- 2 W pliku *drinki.html* zmodyfikuj atrybuty **src**, dopisując przed nazwami plików ścieżkę `../obrazy/`.
- 3 Zapisz oba pliki, a następnie wyświetl plik *salon.html* w przeglądarce. Nawigacja pomiędzy poszczególnymi stronami powinna działać poprawnie, a wszystkie obrazy powinny być widoczne.



PS. Jeśli masz jakiegokolwiek problemy, to poprawnie działającą wersję strony salonu Head First możesz znaleźć w przykładach dołączonych do książki, w katalogu `rozdzial2\salon_skonczony`. Dokładnie porównaj swoje strony z ich odpowiednikami dołączonymi do książki.

Udało Ci się! Teraz nasza witryna jest dobrze zorganizowana, a wszystkie łącza działają poprawnie. Czas, aby to uczcić. Przyłącz się do nas i wypij Zieloną herbatę orzeźwiającą.

A później zabierzemy się za dalsze usprawnianie i rozbudowę naszej witryny.





### KLUCZOWE ZAGADNIENIA

- Kiedy chcesz połączyć dwie strony, użyj do tego celu elementu **<a>**.
- Strona docelowa łącza jest określana przy użyciu atrybutu **href** elementu **<a>**.
- Zawartość elementu **<a>** stanie się etykietą łącza. To właśnie ta etykieta jest wyświetlana na stronie WWW. Domyślnie jest ona podkreślana, co oznacza, że można ją kliknąć.
- Etykietę łącza może stanowić zarówno tekst, jak i obrazy.
- Po kliknięciu łącza przeglądarka wyświetla stronę WWW podaną w atrybucie **href**.
- Łącza mogą wskazywać na pliki znajdujące się zarówno w tym samym, jak i w zupełnie innych katalogach.
- Ścieżka względna jest łączem wskazującym na pliki znajdujące się w tej samej witrynie, przy czym jest ona określana względem położenia strony początkowej. Podobnie jak na mapie także w tym przypadku droga do celu jest określana względem punktu początkowego.
- Użyj symbolu **..**, aby utworzyć łącze z plikiem umieszczonym w katalogu nadrzędnym względem katalogu, w jakim znajduje się strona zawierająca łącze.
- Symbol **../** oznacza „katalog nadrzędny”.
- Pamiętaj, że poszczególne elementy ścieżki należy oddzielać od siebie za pomocą znaku ukośnika (**/**).
- Jeśli ścieżka do obrazu nie będzie poprawna, to zamiast właściwego obrazu przeglądarka wyświetli ikonę przełamanego obrazka.
- Określając nazwy plików i katalogów wchodzących w skład witryny, należy pamiętać, by nie stosować w nich znaków odstępu.
- Odpowiednią organizację witryny należy opracować i zastosować na możliwie wczesnych etapach jej tworzenia; dzięki temu później, kiedy witryna zostanie rozbudowana, nie będziesz musiał poprawiać dziesiątek łączy.
- Istnieje wiele sposobów organizowania witryn WWW. To, który z nich zastosujesz, zależy wyłącznie od Ciebie.

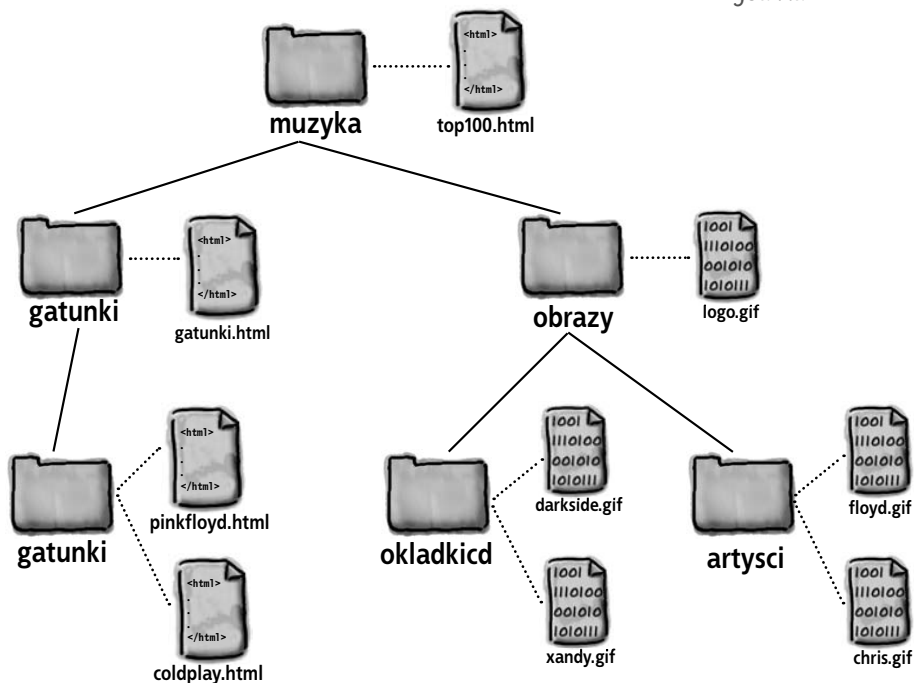


## Wielki konkurs względności

Oto dajemy Ci szansę wypróbowania swoich umiejętności określania ścieżek względnych. Stworzyliśmy witrynę prezentującą 100 najlepszych albumów muzycznych, która jest umieszczona w katalogu *muzyka*. W tym katalogu znajdziesz pliki HTML, inne katalogi oraz obrazy. Twoim zadaniem jest określenie ścieżek względnych, które pozwolą nam na połączenie naszych stron z innymi stronami i plikami wchodzącymi w skład witryny.

Na tej stronie przedstawiliśmy strukturę witryny, na następnej znajdziesz natomiast ćwiczenia, które sprawdzą Twoje umiejętności. Twoim zadaniem jest określenie poprawnej względnej ścieżki dla każdej pary plików: strona początkowa — strona docelowa. Jeśli uda Ci się wykonać to zadanie, to będziesz prawdziwym mistrzem ścieżek.

*Określając ścieżki, możesz rysować je bezpośrednio na tym rysunku.*



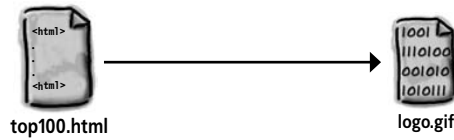
Czas zacząć test.  
Uwaga... Gotów... Pisz!

*Przykład*

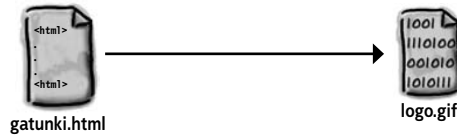


Plik `top100.html` znajduje się w katalogu `muzyka`, a zatem, aby dotrzeć do pliku `gatunki.html`, musimy przejść jeden poziom niżej — właśnie do katalogu `gatunki`.

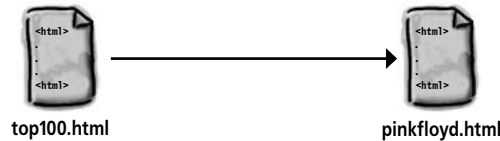
*Runda pierwsza*



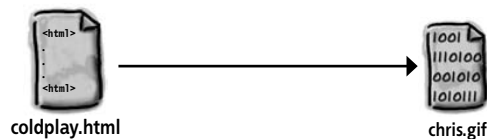
*Runda druga*



*Runda trzecia*



*Runda dodatkowa*





## Rozwiązania ćwiczeń

```

<html>
  <head>
    <title>Firmowe napoje salonu Head First</title>
  </head>
  <body>
    <h1>Nasze napoje firmowe</h1>

    <h2>Zielona herbata orzeźwiająca</h2>
    <p>
      
      Szklanka pełna witamin i minerałów. Ten napój
      to źródło zdrowia, jakie zapewnia zielona herbata
      oraz mieszanka płatków rumianku i korzenia imbiru.
    </p>

    <h2>Koncentracja malinowej ochłody</h2>
    <p>
      
      Ten orzeźwiający napój, stanowiący połączenie soku malinowego
      z trawą cytrynową, skórką cytrynową oraz owocami głogu sprawi,
      że Twój umysł stanie się bystry i chłonny.
    </p>

    <h2>Niebiańska borówka</h2>
    <p>
      
      Wstrząśnięta esencja jagód i wiśni, zmieszana z herbatą
      z czarnego bzu błyskawicznie, sprawi, że poczujesz błogą
      rozkosz.
    </p>

    <h2>Eksplozja żurawinowego antyutleniacza</h2>
    <p>
      
      Otrzeźwiej, wdychając aromaty żurawin i hibiskusa zawarte
      w lekkim napoju pełnym witaminy C.
    </p>
    <p>
      <a href="salon.html">Powrót na stronę salonu</a>
    </p>
  </body>
</html>

```

Umieściliśmy łącze w osobnym akapicie, aby zachować porządek i przejrzystość strony. Więcej informacji na ten temat podamy w następnym rozdziale.



Oto nowy element <a>, stanowiący łącze prowadzące do głównej strony salonu.



## Rozwiązania ćwiczeń



Etykieta	Strona docelowa	Kod HTML
Gorący czy nie?	<i>goracy.html</i>	<code>&lt;a href="goracy.html"&gt;Gorący czy nie?&lt;/a&gt;</code>
Życiorys	<i>cv.html</i>	<code>&lt;a href="cv.html"&gt;Życiorys&lt;/a&gt;</code>
Cukiereczek	<i>cukiereczek.html</i>	<code>&lt;a href=" <u>cukiereczek.html</u> " &gt;Cukiereczek&lt;/a&gt;</code>
Oto ja w moim mini	<i>mini-cooper.html</i>	<code>&lt;a href="mini-cooper.html"&gt;Oto ja w moim mini&lt;/a&gt;</code>
Zagrajmy	<i>szczesciarz.html</i>	<code>&lt;a href="szczesciarz.html"&gt; <u>Zagrajmy</u> &lt;/a&gt;</code>



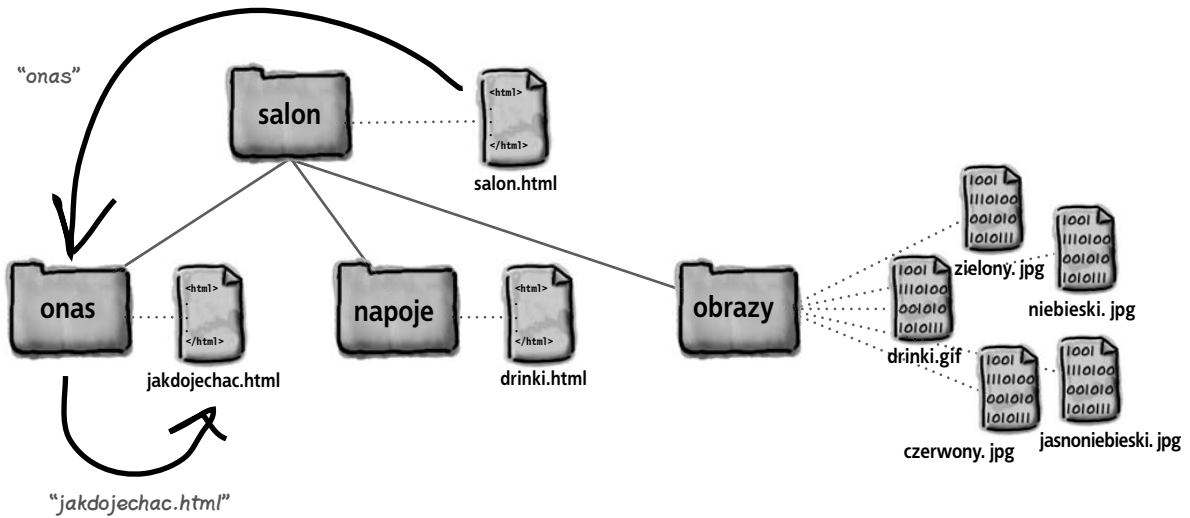
## Zaostrz ołówek



### Rozwiązanie

Określ względną ścieżkę od pliku *salon.html* do pliku *jakdojehac.html*. Kiedy już to zrobisz, uzupełnij podany poniżej element `<a>`.

A oto i rozwiązanie ćwiczenia. Czy zmieniłeś oba elementy `<a>` umieszczone w kodzie strony *salon.html*?



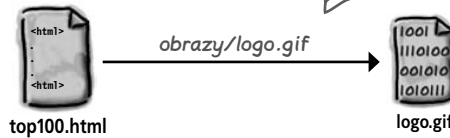
```
<a href= "onas/jakdojehac.html" >opisem dojazdu</a>
```

TU WPISZ SWOJĄ ODPOWIEDŹ



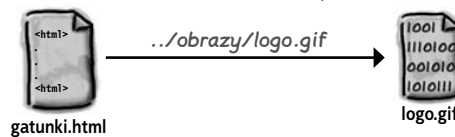
## Rozwiązanie wielkiego konkursu względności

### Runda pierwsza



Plik top100.html znajduje się w katalogu muzyka, a zatem, aby dotrzeć do pliku gatunki.html, musimy przesunąć się o jeden poziom niżej — właśnie do katalogu gatunki.

### Runda druga



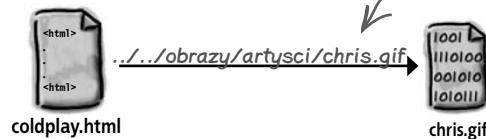
Strona gatunki.html znajduje się na niższym poziomie, w katalogu gatunki, a zatem, aby dotrzeć z niej do pliku logo.gif, musieliśmy najpierw wyjść na wyższy poziom do katalogu muzyka, a następnie wejść do katalogu obrazy.

### Runda trzecia



Ze strony top100.html wchodzimy do katalogu gatunki, następnie do katalogu rock, a w nim możemy znaleźć plik pinkfloyd.html.

### Runda dodatkowa



To była najtrudniejsza ścieżka. Zaczynając do strony coldplay.html znajdującej się w katalogu rock, musieliśmy przesunąć się o DWA poziomy do góry, aby dostać się do katalogu muzyka; następnie musieliśmy wejść do katalogu obrazy, następnie do katalogu artyści i dopiero w nim mogliśmy znaleźć poszukiwany plik chris.gif. O rany!