

» Idź do

- Spis treści
- Przykładowy rozdział

» 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

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2008

XHTML, CSS i JavaScript. Pierwsza pomoc

Autorzy: [Maria Sokół](#), [Radosław Sokół](#)

ISBN: 978-83-246-1071-6

Format: A5, stron: 100



- Początki – tworzenie szkieletu strony WWW i struktury dokumentu HTML
- Wygląd strony – kodowanie znaków, krój pisma i kaskadowe arkusze stylów
- Elementy witryny – listy, tabele, obrazy, odnośniki
- Interaktywność – formularze, bazy danych, historia działań użytkownika

Masz świetne pomysły i chciałbyś pokazać je światu? A może założyłeś firmę i chcesz się zareklamować? Najlepszą współczesną wizytówką jest interesująca, funkcjonalna strona internetowa. Jeśli chcesz idealnie dopasować ją do swoich potrzeb, najlepiej byłoby zająć się nią osobiście. Jak jednak stworzyć taką stronę? Jak zapewnić jej dobre działanie, intuicyjność obsługi i właściwą strukturę? O co chodzi z tymi wszystkimi znacznikami? Co to jest CSS? Jak rozmieścić poszczególne elementy i ułatwić poruszanie się po witrynie? I czy da się ją połączyć z bazą danych? Nie martw się – spieszmy z pierwszą pomocą. Na te oraz wiele innych technicznych pytań szybko i jasno odpowie Ci ta właśnie książka!

- Projektowanie – tworzenie szkieletu strony WWW i struktury dokumentu HTML
- Czynności podstawowe – używanie znaczników, kodowanie znaków
- Formatowanie – definiowanie położenia, rozmiaru i koloru tekstu, ustawianie koloru strony
- Używanie kaskadowych arkuszy stylów (CSS)
- Dodawanie elementów – listy, tabele, odnośniki zewnętrzne i pocztowe
- Wstawianie obrazów – zmiana rozmiaru, ustawianie względem tekstu, pozycja na stronie
- Rola pakietu XAMPP i umieszczanie strony na serwerze
- Kontakt z użytkownikiem – formularze, wprowadzanie danych, historia odwiedzin
- Bazy danych – połączenie ze stroną, operacje na danych, prawa dostępu
- Wykorzystywanie technologii AJAX na stronach WWW

Oto recepta na szybkie stworzenie własnej strony WWW!

Spis treści

- Wstęp / 5
- 1. Co to są znaczniki? / 7
- 2. Jak utworzyć podstawowy szkielet strony WWW? / 10
- 3. Gdzie umieścić tytuł strony WWW? / 13
- 4. Co to jest kodowanie znaków i jakie są jego techniki? / 14
- 5. Gdzie ustawić kodowanie polskich znaków w kodzie HTML? / 16
- 6. Jak korzystać z atrybutów tekstu? / 19
- 7. Jak zdefiniować wyrównanie tekstu w akapicie? / 21
- 8. Jak zdefiniować kolor tekstu i tła akapitu? / 24
- 9. Jak zdefiniować krój pisma i rozmiar czcionki? / 27
- 10. Jak ustawić marginesy akapitu tekstu? / 29
- 11. Jak budować strukturę dokumentu HTML za pomocą nagłówków? / 31
- 12. Jak utworzyć na stronie WWW listy wypunktowane i numerowane? / 33
- 13. Jak przygotować prostą stronę WWW? / 36
- 14. Jak definiować kaskadowe arkusze stylów? / 38
- 15. Co to jest odnośnik i jak go skierować do innej strony WWW? / 43
- 16. Jak modyfikować odnośniki za pomocą CSS? / 47
- 17. Jak zdefiniować odnośnik pocztowy? / 49
- 18. Do czego służą kotwice nazwane? / 50
- 19. Jak wstawić obraz na stronę WWW? / 53
- 20. Jak zmienić rozmiary obrazu? / 56
- 21. Jak określić pozycję obrazu na stronie WWW? / 57
- 22. W jaki sposób określić relację obraz – tekst? / 58
- 23. Jak zapewnić płynną zmianę rozmiarów obrazu przy zmianie rozmiarów okna przeglądarki? / 60



24. Jak zdefiniować tabelę w XHTML? / 62
25. Jak wykorzystać tabelę do budowy strony WWW? / 65
26. Jak stworzyć układ strony o postaci tabeli z wykorzystaniem wyłącznie CSS? / 68
27. Jak umieścić na stronie WWW formularz? / 75
28. Jak wyświetlić na ekranie komunikat z wykorzystaniem JavaScript? / 81
29. Jak wyświetlić element w oknie pop-up za pomocą JavaScript? / 85
30. Jak przygotować galerię obrazów? / 87
31. Jak zdefiniować menu rozwijane za pomocą JavaScript? / 90



27. Jak umieścić na stronie WWW formularz?

Formularze pozwalają autorowi strony zbierać dane od osób odwiedzających jego serwis WWW i automatyzować kontakty. Mają postać elektronicznej ankiety, którą wypełnia się wprost na stronie.

Formularz definiujemy za pomocą elementu `form` z odpowiednimi atrybutami, w którym umieszczamy pola formularza:

```
<form action="plik_docelowy" method="post">  
  zawartość formularza  
</form>
```

Pamiętaj, aby wszelkie zdania w formularzu, pola, zestawy opcji itp. obejmować jakimiś pojemnikami, na przykład znacznikami akapitu `<p>` czy bloku `<div>`.

W tym przykładzie przygotowujemy formularz, który pozwoli na wysyłanie za pomocą poczty elektronicznej prowadzanych przez użytkownika informacji. W związku z tym w definicji formularza powinniśmy umieścić specjalne polecenie wysyłania formularza na wskazany adres:

```
action="mailto:autor_strony@adres.pl".
```

Należy również określić sposób komunikowania się przeglądarki użytkownika z serwerem, a więc wskazać w kodzie, czy przesyła ona jakieś informacje, czy też je pobiera. Do wyboru mamy dwie metody — `post` lub `get` — zastosujemy metodę `post`.

Aby zebrane z formularza dane były w czytelnej postaci, powinniśmy zastosować parametr `enctype="text/plain"`. Tak więc znacznik `<form>` ma teraz postać:

```
<form enctype="text/plain" action="mailto:autor_strony@adres.  
pl?subject=Ankieta" method="post">
```

Formularz może zawierać różnego typu pola, w które będą wprowadzane dane — na przykład takie jak imię i nazwisko, nazwa produktu, informacje o użytkowniku itp. Pola te są podzielone na kilka podstawowych grup:

- ▶ Pole `input`, z dodatkowymi parametrami `type`, `name` i `value`, służy do tworzenia pól, w których czytelnik strony może wpisać informację lub wybrać jakąś opcję.



Oto przykład:

```
<input type="text" name="nazwa_pola" value="wartość początkowa"
size="rozmiar pola w znakach" maxlength="maksymalny rozmiar
pola">
```

lub

```
<p><input type="checkbox" name="wybor" value="tak"> Tak</p>
```

- Pole **select** służy do tworzenia rozwijalnych list z opcjami, spośród których czytelnik wybiera interesującą go pozycję:

```
<p><strong>Wybierz produkt:</strong></p>
<div>
<select name="warzywa" size="3">
<option> Avocado
<option> Cebula
<option> Ogórek
<option> Marchew
<option> Seler
</select>
</div>
```

- **textarea** jest poleceniem pozwalającym tworzyć większe pole tekstowe z przeznaczeniem na dłuższy komentarz czytelnika:

```
<form action="mailto:autor_strony@adres.pl">
<p><b>Wpisz swoje uwagi:</b></p>
<p><textarea name="Uwagi" rows="5" cols="55"></textarea></p>
</form>
```



Dodatkowo możemy wydzielić określone pola poprzez zastosowanie do nich grupującego obramowania. Służy do tego element `fieldset`:

```
<form>
  <fieldset>
    zawartość
  </fieldset>
  <fieldset>
    zawartość
  </fieldset>
</form>
```

Przydatny będzie także opis grupy pól — wprowadza go element `legend`, na przykład:

```
<legend>Formularz - stosowane przeglądarki</legend>
```

Do formularza trzeba jeszcze dodać przycisk, który pozwoli wysłać informacje podane przez czytelnika strony do jej autora. Służy do tego polecenie `input` z atrybutem `type="submit"`. Gdy użytkownik kliknie ten przycisk, wszystkie podane przez niego informacje (ciągle zakładamy, że posługujemy się tu funkcją poczty elektronicznej), jak zaznaczone opcje, wypełnione pola tekstowe czy wybrane pozycje na liście, zostaną przesłane w liście do autora:

```
<input type="submit" value="Wyślij do nas informacje">
```

Teraz czas na zastosowanie podanych tu informacji w praktyce.



Aby umieścić na stronie WWW formularz:

1. Otwórz w Notatniku podstawowy szablon strony WWW.
2. Umieść w kodzie strony WWW element **form**, a w nim zdefiniuj cztery obramowania **fieldset** wydzielające zestawy danych określonego typu:
 - ▶ Pierwszy zestaw zawiera pola tekstowe, w które użytkownik będzie wpisywał swoje dane personalne.
 - ▶ Zestaw drugi zawiera przyciski radiowe pozwalające wybrać płeć.
 - ▶ Zestaw trzeci to lista przewijana, na której użytkownik będzie wskazywał swój wybór.
 - ▶ Czwarty zestaw to pola wyboru.
3. Na końcu formularza umieść kod definiujący przycisk wysyłania formularza na wskazany adres.
4. Pełny kod formularza został przedstawiony na listingu 27.1.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:
lang="pl" lang="pl">
  <head>
    <title>FORMULARZ</title>
    <meta http-equiv="Content-Type" content="text/
html; charset=utf-8" />
  </head>
  <body>
    <form enctype="text/plain" action="mailto:mary@
mary.gliwice.pl?subject=Formularz" method="post">
      <fieldset>
        <legend>Formularz - stosowane przeglądarki</
legend>
        <p><strong>Imię</strong>: <input type="text"
name="Imie" value="Jan" size="30" maxlength="40"
style="position:absolute; left:150px; background-
color:cornsilk; font-style: italic; color:gray"
 /></p>
        <p><strong>Nazwisko</strong>: <input type="text"
name="Nazwisko" value="Kowalski" size="30"
maxlength="40" style="position:absolute;
left:150px; background-color:cornsilk; font-
style: italic; color:gray" /></p>
        <p><strong>Zawód</strong>: <input type="text"
name="Zawod" value="student" size="30"
maxlength="40" style="position:absolute;
left:150px; background-color:cornsilk; font-
style: italic; color:gray" /></p>
      </fieldset>
      <fieldset>
        <p><b>Podaj swoją płeć</b></p>
        <p><input type="radio" name="Plec" value="kobieta"
checked="checked" /> <strong>Kobieta</strong>
<br />
```

Listing 27.1. Formularz

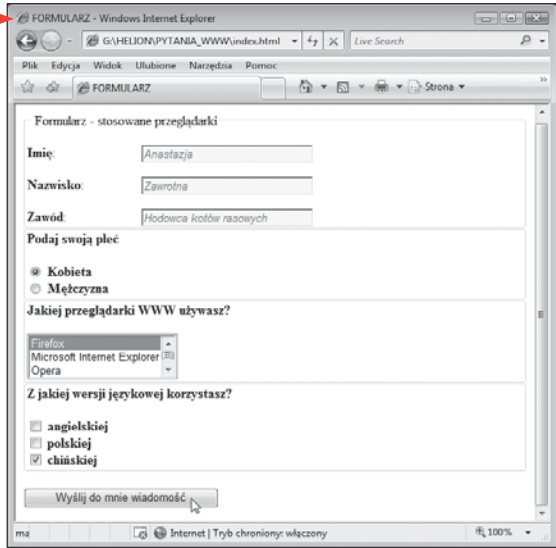


```
<input type="radio" name="Plec" value="mężczyzna"
 /> <strong>Mężczyzna</strong></p>
</fieldset>

<fieldset>
<p><b>Jakiej przeglądarki WWW używasz?</b></p>
<div><select name="Przeglądarka" size="3">
<option>Firefox</option>
<option>Microsoft Internet Explorer</option>
<option>Opera</option>
<option>Safari</option>
<option>Innej</option>
</select></div>
</fieldset>

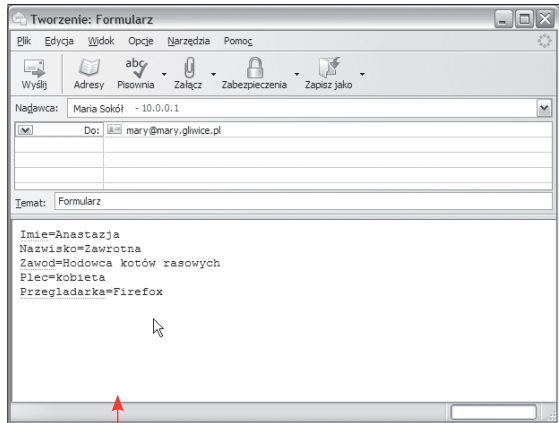
<fieldset>
<p><b>Z jakiej wersji językowej korzystasz?</b></p>
<div><input type="checkbox" name="Jezyk"
value="angielski" /> <strong>angielskiej</strong><br />
<input type="checkbox" name="Jezyk" value="polski"
 /> <strong>polskiej</strong><br />
<input type="checkbox" name="Jezyk"
value="chiński" /> <strong>chińskiej</strong><br />
</div>
</fieldset>

<br />
<input type="submit" value="Wyślij do mnie
wiadomość" />
</form>
</body>
</html>
```

Rysunek 27.1. Oto formularz, który wygeneruje wiadomość e-mail i umieści w jej treści dane podane w polach formularza

- Przeanalizuj kod z listingu 27.1. Zapisz plik zawierający formularz pod nową nazwą i wyświetl w przeglądarce (rysunek 27.1).
- Przetestuj funkcjonalność formularza. Wypełnij jego pola i kliknij przycisk wysyłania danych. Otworzy się okno wiadomości e-mail, w którym pole treści będzie zawierać rekordy danych (rysunek 27.2).



Rysunek 27.2. Formularz wygenerował rekordy danych — zostaną one przesłane na adres autora strony WWW



JavaScript to język programowania, który pozwala rozbudowywać funkcjonalność stron WWW. Możesz w tym języku zaprogramować kalkulator lub prostą grę, ale także całkiem poważne aplikacje, które kontrolują typ przeglądarki czy ciasteczka.

Zanim zabierzemy się do pracy — kilka praktycznych uwag. Ponieważ JavaScript to nie XHTML, konieczne jest poinformowanie przeglądarki, że w kodzie strony WWW zawarty jest program JavaScript. Służy do tego element `script`. Element `script` może być umieszczony równie dobrze w sekcji `head`, jak i w sekcji `body`, ale raczej zdecyduj się na tę pierwszą opcję. W nim wpisujemy polecenia kodu JavaScript. Oto podstawowe zasady obowiązujące w obszarze tego elementu:

- ▶ Każdy wiersz kodu JavaScript kończy się średnikiem. Natrafiając na średnik, przeglądarka wie, że wykonała wiersz kodu i że może przejść do kolejnego.
- ▶ Tekst jest umieszczany ZAWSZE w cudzysłowach " ". Jeśli zapomnisz cudzysłowów, tekst zostanie potraktowany jako zmienna.
- ▶ Wielkie litery i małe litery nie są identyczne. Jeśli więc zdefiniujesz zmienną `ZMIENNA`, to nie będzie ona tożsama ze zmienną o nazwie `zmienna`. Pamiętaj o tej zasadzie jako o możliwym źródle błędów.

Zacznijmy od umieszczenia w kodzie strony WWW polecenia JavaScript otwierającego okno komunikatu. W rzeczywistości są trzy typy takich okien otwieranych instrukcjami `alert`, `prompt` i `confirm` — wykorzystamy je wszystkie do stworzenia prostego systemu komunikacji z użytkownikiem. Oto podstawowa składnia:

- ▶ `alert ("Dowolny tekst.");` — instrukcja wyświetla okno dialogowe zawierające tekst, który umieścisz między znakami cudzysłowu, oraz przycisk `OK`.
- ▶ `confirm ("Dowolny tekst?");` — instrukcja wyświetla okno dialogowe z dwoma przyciskami, `OK` i `Cancel`. Pierwszy przycisk odpowiada wartości `TRUE` (prawda), drugi wartości `FALSE` (fałsz).
- ▶ `prompt ("Dowolny tekst ?", "");` — ta instrukcja pobiera dane od użytkownika. W wyświetlonym oknie dialogowym pojawi się pole tekstowe, w które użytkownik wprowadzi dane.

28. Jak wyświetlić na ekranie komunikat z wykorzystaniem JavaScript?

Aby za pomocą kodu JavaScript wyświetlić na ekranie okno komunikatu:

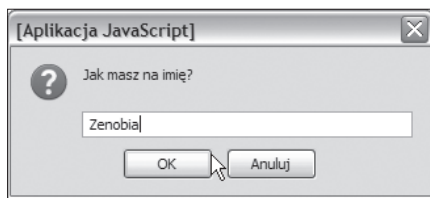
1. Otwórz w Notatniku plik zawierający podstawowy szablon dokumentu HTML.
2. W sekcji **head** dokumentu HTML (lub w sekcji **body**) umieść element **script** o następującej postaci:

```
<script type="text/javascript">
  <!--
  var imie = prompt ("Jak masz na imię?","");
  alert ("Cześć, " + imie);
  var jak_masz = confirm ("Jak się masz?");
  if (jak_masz)
    alert ("Super! Ja także.");
  else
    alert ("Oj, to niedobrze :( ");
  // -->
</script>
```

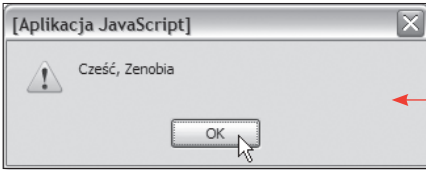
Znaczniki **<script>** i **</script>** informują przeglądarkę, gdzie jest początek i koniec kodu JavaScript.

Aby zabezpieczyć się przed potraktowaniem kodu JavaScript jako zwykłego tekstu, wszystkie polecenia umieszczone w obrębie elementu **script** zostały zawarte między znacznikami komentarza **<!--, // -->**.

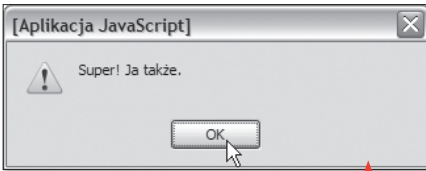
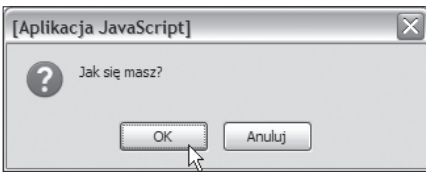
var definiuje zmienną — pierwszej nadajemy nazwę **imie** (możesz zdefiniować dowolną nazwę, ale nie używaj polskich znaków diakrytycznych; nie zaczynaj też nazwy od cyfry). Zmienna ta będzie przechowywać tymczasowo wartość odpowiadającą odpowiedzi użytkownika. W przykładowym kodzie wartością zmiennej jest tekst wprowadzony przez użytkownika w oknie dialogowym wywołanym instrukcją **prompt** ("Jak masz na imię?,"") (rysunek 28.1).



Rysunek 28.1. Okno wywołane instrukcją **prompt**. Jeśli naciśniesz OK, wprowadzony tekst zostanie wykorzystany przez skrypt w kolejnym oknie



Rysunek 28.2. Okno wywołane instrukcją `alert`. Za pomocą tej instrukcji możesz wyświetlać dowolny komunikat przeznaczony dla użytkownika

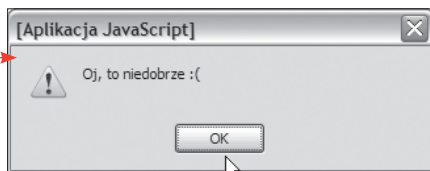


Rysunek 28.3. Okno wywołane instrukcją `confirm`. Odpowiedź programu zależy od tego, który z przycisków zostanie naciśnięty przez użytkownika — to instrukcja warunkowa `if` wybierze właściwą odpowiedź

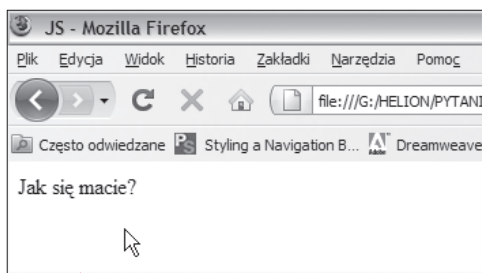
Wprowadzony tekst zostanie wykorzystany w oknie komunikatu. Jest to możliwe dzięki zastosowaniu operatora dodawania `+`, za którego pomocą do tekstu komunikatu okna `alert` zostanie dodana wartość zmiennej `imie`: `alert ("Cześć, " + imie);`. Efekt pokazano na rysunku 28.2.

W kolejnym wierszu kodu definiowana jest zmienna `jak_masz`, która ma przechowywać wartość odpowiedzi użytkownika wprowadzonej w oknie `confirm`: `var jak_masz = confirm ("Jak się masz?");`. Jeśli użytkownik naciśnie przycisk `OK`, zmienna zyska wartość `TRUE`, jeśli naciśnie przycisk `Cancel`, zmienna będzie miała wartość `FALSE` — skorzystamy więc z instrukcji warunkowej `if...else`, aby zareagować na poczynania użytkownika (rysunek 28.3).

Instrukcja warunkowa `if ... else` kieruje reakcją na poczynania użytkownika. Jeśli kliknie on przycisk `OK`, zmienna `jak_masz` przyjmuje wartość `TRUE` i wyświetlane jest okno z komunikatem określonym w instrukcji `alert ("Super! Ja także.");`



Rysunek 28.4. To okno komunikatu jest efektem realizacji części else instrukcji warunkowej if



Rysunek 28.5. Efekt działania instrukcji `document.write ()` — pozwala ona wyświetlać dowolny tekst bezpośrednio na ekranie. Spośród różnorodnych zastosowań metody `document.write` warto wyróżnić możliwość dopisywania kodu HTML do strony WWW

Jeśli użytkownik kliknie przycisk *Cancel*, zmienna `jak_masz` przyjmie wartość `FALSE` i realizowana będzie część `else` instrukcji warunkowej — instrukcja `alert` wyświetli komunikat *Oj, to niedobrze :(* (rysunek 28.4).

3. Jeśli chcesz wyświetlić na ekranie wartość którejś ze zmiennych, na przykład wartość zmiennej `imie`, skorzystaj z metody `document.write` (*nazwa_zmiennej*).

`document.write ()` to metoda JavaScript informująca przeglądarkę, że powinna ona wyświetlić na ekranie monitora wszelką zawartość określoną w nawiasach — może to być także dowolny tekst. Jeśli więc chcesz wyświetlić taki tekst, umieść go w cudzysłowach, na przykład: `document.write ("Jak się macie?");` (rysunek 28.5).



Czasem trzeba otworzyć małe okienko o określonej wielkości z krótką informacją, zdjęciem czy ankietą. W takiej sytuacji można skorzystać ze zdarzeń JavaScript. Zdarzenia umożliwiają uruchomienie dowolnego fragmentu skryptu JavaScript, gdy wystąpi wskazane zdarzenie, na przykład przycisk zostanie kliknięty.

Oto przykłady zdarzeń oraz opis sytuacji, w których zachodzą:

- ▶ **onblur** — przy przełączeniu do innego elementu.
- ▶ **onclick** — przy kliknięciu elementu.
- ▶ **onchange** — przy zmianie zawartości pola.
- ▶ **onfocus** — przy uaktywnieniu danego elementu.
- ▶ **onmouseover** — przy umieszczeniu nad elementem wskaźnika myszy.
- ▶ **onunload** — przy opuszczeniu strony WWW.

Skorzystamy ze zdarzenia **onclick**, aby wyświetlić okno pop-up, a w nim nową zawartość. Może to być zawartość serwisu, na przykład zdjęcie, lub zawartość innej strony WWW.

```
<body>
<a href="#" onclick="window.open('http://helion.
pl', 'Okno_01', 'height=350, width=300');">Otwórz
stronę WWW w małym okienku</a>
<br />
<a href="#" onclick="window.open('obrazy/
pimpek1.jpg', 'Okno_02', 'height=250,
width=200');">Wyświetl zdjęcie w okienku</a></
body>
</body>
```

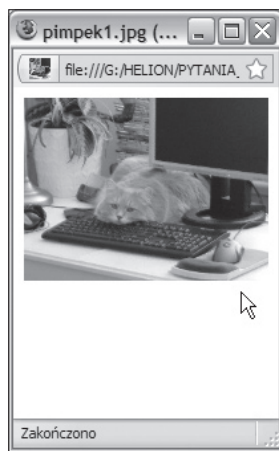
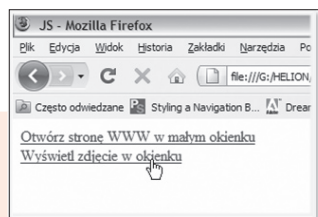
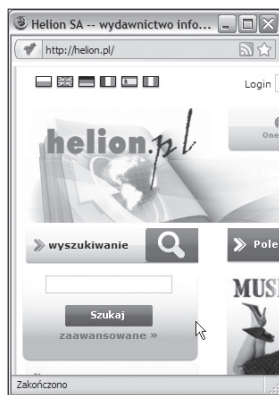
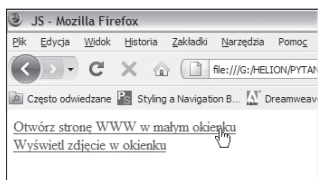
Listing 29.1. Zastosowanie zdarzenia onclick do otwierania okna pop-up

29. Jak wyświetlić element w oknie pop-up za pomocą JavaScript?

Aby wyświetlić okno pop-up:

1. Umieść w sekcji **body** dokumentu elementy **a** — są to elementy definiujące połączenia do zewnętrznej strony WWW i do pliku JPG serwisu WWW. Oto postać sekcji **body** (listing 29.1):





2. W wyniku kliknięcia pierwszego połączenia myszą, a więc wystąpienia zdarzenia `onclick`, zostanie otworzona strona o adresie podanym jako pierwszy argument metody `window.open()`. Drugim argumentem jest nazwa okienka pop-up, a kolejne argumenty podają jego wysokość i szerokość. Parametrów może być więcej — można na przykład włączyć przewijanie lub określić miejsce wyświetlenia okienka na ekranie. Kliknięcie drugiego połączenia spowoduje wyświetlenie obrazu w oknie przeglądarki. Rysunek 29.1 pokazuje efekty działania kodu.

Rysunek 29.1. Połączenia do strony WWW i pliku serwisu funkcjonują dzięki zdarzeniu `onclick` — w wyniku kliknięcia myszą, a więc wystąpienia zdarzenia `onclick`, w oknie pop-up zostanie otworzona zawartość o adresie podanym jako pierwszy argument metody `open()`