

# 7

## FORMULARZE

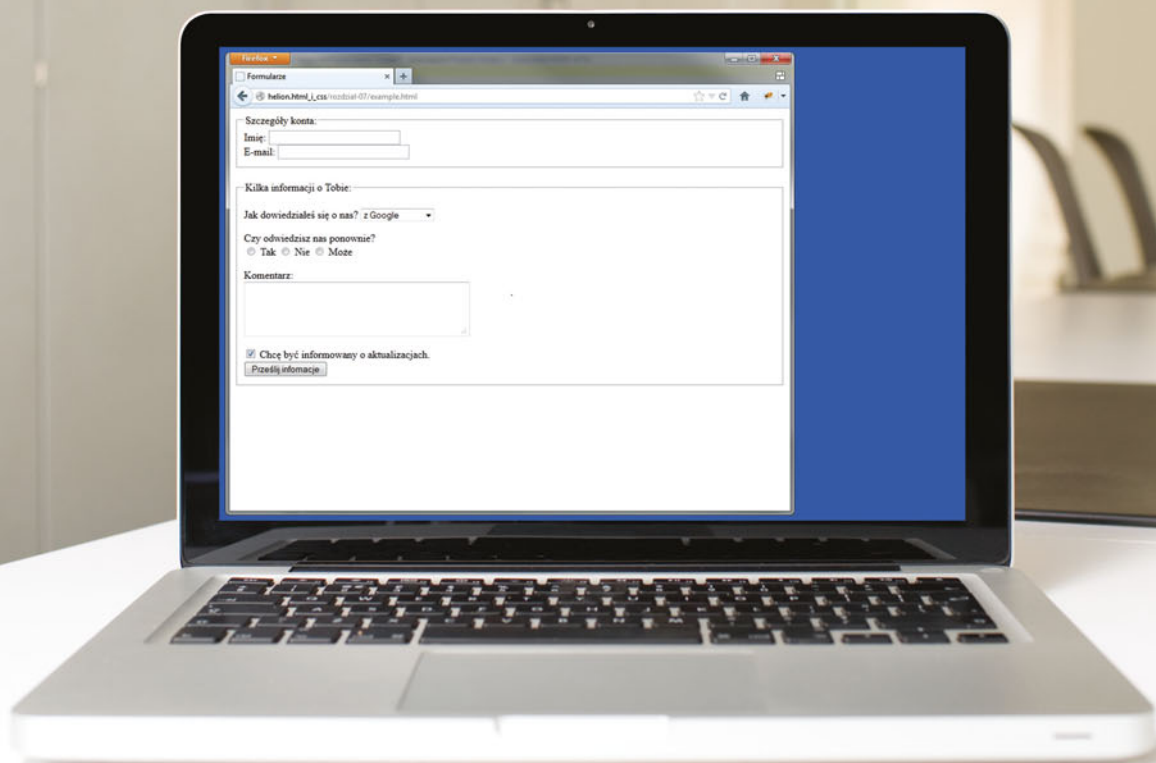
- ▶ Jak zbierać informacje od odwiedzających?
- ▶ Różne rodzaje pól formularzy.
- ▶ Nowe pola formularzy w HTML5.

Tradycyjnie termin „formularz” oznaczał drukowany dokument zawierający puste obszary, w których można było wpisywać informacje.

Język HTML zapożyczył pojęcie formularza i używa go do określania różnego typu elementów pozwalających na pobieranie informacji od osób oglądających witrynę.

Niezależnie od tego, czy dodajemy proste pole wyszukiwania, czy tworzymy złożoną aplikację ubezpieczeniową, formularze HTML udostępniają zbiór elementów umożliwiających pobieranie danych od użytkowników. W tym rozdziale zostaną opisane:

- Sposoby tworzenia formularzy na stronie WWW.
- Różne narzędzia służące do gromadzenia informacji.
- Nowe pola formularzy dostępne w HTML5.

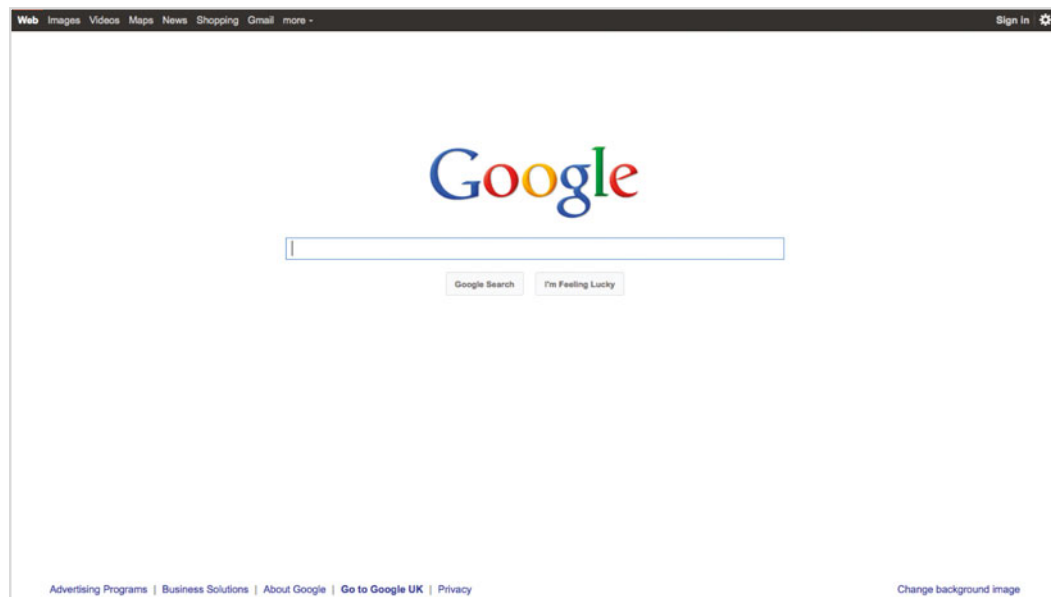


# DLACZEGO FORMULARZE?

Prawdopodobnie najbardziej znanym formularzem w sieci jest pole tekstowe służące do wyszukiwania, umieszczone na środku głównej strony wyszukiwarki Google.

Oprócz zapewnienia użytkownikom możliwości wyszukiwania formularze pozwalają na wykonywanie innych czynności. Korzystamy z nich, kiedy rejestrujemy się

w nowej witrynie, kupujemy coś w sklepie internetowym, subskrybujemy jakiś biuletyn lub zapisujemy się do listy dyskusyjnej.



# POLA FORMULARZY

Istnieje kilka rodzajów pól formularzy, których można używać do pobierania informacji od osób odwiedzających witrynę.

## DODAWANIE TEKSTU:

### Pola tekstowe (jednowierszowe)

Używane do wprowadzania jednego wiersza tekstu, zawierającego np. adres poczty elektronicznej lub imię.

### Pola hasła

Podobne do jednowierszowych pól tekstowych, tyle że maskują wprowadzane znaki.

### Wielowierszowe pola tekstowe

Służą do tworzenia większych pól tekstowych używanych do wprowadzania długich wiadomości lub komentarzy.

## DOKONYWANIE WYBORÓW:

### Przyciski opcji

Stosowane, gdy użytkownik musi wybrać jedną z kilku dostępnych opcji.

☒ Rock ☐ Pop ☐ Jazz

### Pola wyboru

Stosowane, gdy użytkownik może zaznaczyć lub usunąć zaznaczenie jednej lub kilku opcji.

☒ iTunes ☐ Last.fm ☐ Spotify

### Listy rozwijane

Stosowane, gdy użytkownik musi wybrać jedną spośród opcji dostępnych na liście.

## PRZESYŁANIE FORMULARZY:

### Przycisk przesyłający

Służy do przesyłania danych z formularza na serwer.

### Przycisk graficzny

Podobny do przycisku przesyłającego, można w nim stosować obrazki.

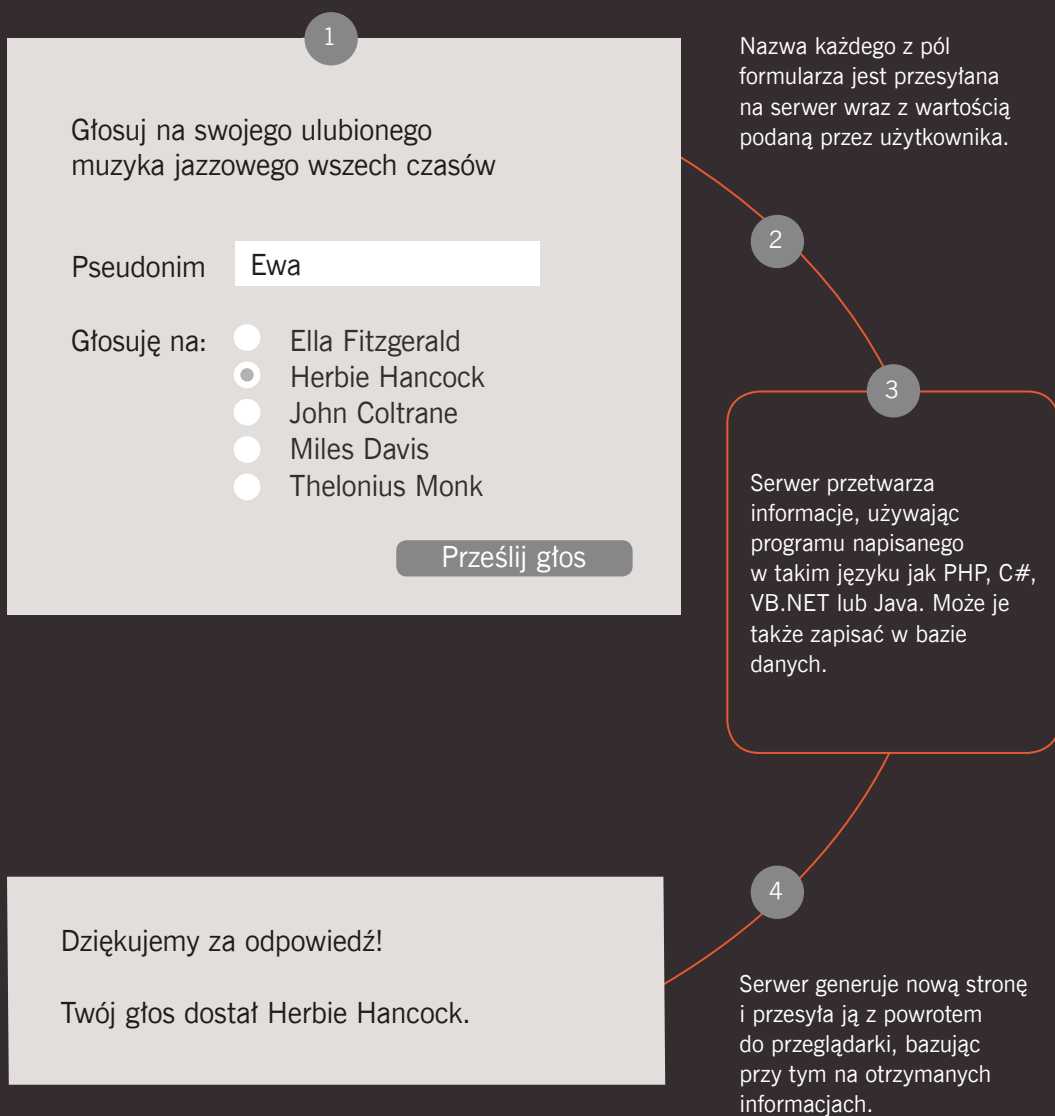
## PRZESYŁANIE PLIKÓW:

### Przesyłanie plików

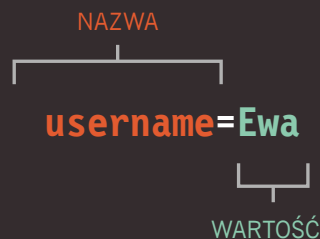
To pole pozwala użytkownikom przysyłać pliki (np. obrazki) na serwer WWW.

# JAK DZIAŁAJĄ FORMULARZE?

Użytkownik wypełnia formularz, a następnie naciska przycisk, by przesłać podane w nim informacje na serwer.



W formularzu można umieścić kilka pól, z których każde będzie zawierać inne informacje. Serwer musi wiedzieć, które z wprowadzonych danych są powiązane z danym polem formularza.



Aby odróżniać różne rodzaje wprowadzanych danych, informacje są przesyłane z przeglądarki na serwer w formie par nazwa – wartość. W przedstawionym przykładzie w formularzu poproszono użytkownika o podanie imienia oraz wybór ulubionego muzyka jazzowego. Poniżej przedstawione zostały pary nazwa – wartość przysyłane na serwer.

**username=Ewa**

Jeśli pole formularza pozwala na wpisanie informacji, to jego wartością będzie dowolny tekst umieszczony w nim przez użytkownika.

**vote=Herbie**

Jeśli pole pozwala na wybranie jednej spośród kilku konkretnych wartości (jak jest w przypadku przycisków opcji lub list rozwijanych), to autor strony sam poda kod automatycznie określający wartość danej opcji.

Nigdy nie zmieniamy nazwy pola formularza na stronie, jeśli nie jesteśmy pewni, że kod obsługujący formularz po stronie serwera prawidłowo obsłuży nową nazwę.

# STRUKTURA FORMULARZY

## <form>

Wszystkie pola formularzy są umieszczane wewnątrz elementu <form>. W tym elemencie zawsze należy określić wartość atrybutu action; oprócz tego zazwyczaj podawane są wartości atrybutów method oraz id.

### action

Każdy element <form> wymaga określenia wartości atrybutu action. Jest nią adres URL strony na serwerze, do której w momencie wysłania formularza należy przekazać podane w nim informacje.

### method

Formularze można wysyłać, używając jednej z dwóch metod: get albo post.

W przypadku stosowania metody get wartości pól formularza są dodawane na końcu adresu URL podanego w atrybucie action. Najlepiej nadaje się ona do:

- przesyłania krótkich formularzy (takich jak formularz wyszukiwania);
- formularzy służących jedynie do pobierania danych z serwera (a nie np. przesyłania informacji, które mają być zapisane lub usunięte z bazy danych).

rozdzial-07/form-structure.html

HTML

```
<form action="http://www.jakaswitryna.com/subscribe.php"
      method="get">
  <p>Tu będą umieszczone pola formularza.
</p>
</form>
```

WYNIK

Tu będą umieszczone pola formularza.

W przypadku zastosowania metody post wartości są przesyłane w tzw. nagłówkach HTTP. Jako ogólną zasadę należy przyjąć, że metoda post powinna być użyta, gdy formularz:

- pozwala na przesyłanie plików;
- jest bardzo duży;
- zawiera wrażliwe dane (takie jak hasła);
- powoduje dodanie informacji do bazy danych lub usunięcie informacji z bazy danych.

Jeśli atrybut method nie został podany, przeglądarka prześle formularz, używając metody get.

### id

Atrybut id został dokładniej opisany na stronie 182; jego wartość jest używana do unikalnego identyfikowania elementu formularza i odróżniania go od pozostałych elementów strony (bardzo często stosuje się go w skryptach, np. takich, które sprawdzają, czy w formularzu zostały podane wszystkie niezbędne dane).



# POLA TEKSTOWE

## HTML

rozdzial-07/text-input.html

```
<form action="http://www.jakaswitryna.com/login.php">
  <p>Nazwa użytkownika:
    <input type="text" name="username" size="15"
      maxlength="30" />
  </p>
</form>
```

## WYNIK

Nazwa użytkownika:

## size

Atrybutu `size` nie należy używać w nowych formularzach. Wcześniej był on wykorzystywany w celu określenia długości pola (jego wartość określała liczbę znaków, które miały być widoczne w polu).

Użycie np. wartości 3 spowoduje utworzenie pola tekstowego o takiej szerokości, by były w nim widoczne trzy

znaki (choć w razie potrzeby użytkownik może w nim wpisać więcej znaków).

We wszystkich obecnie tworzonych formularzach szerokość pól należy określać przy użyciu arkuszy stylów CSS. Atrybut `size` został tu opisany tylko dlatego, że można się na niego natknąć podczas przeglądania kodu starych formularzy.

## <input>

Element `<input>` służy do tworzenia kilku różnych rodzajów pól. Rodzaj pola, które zostanie utworzone, zależy od wartości atrybutu `type` tego elementu.

## type="text"

Jeśli atrybut `type` będzie miał wartość `text`, zostanie utworzone pole tekstowe.

## name

Kiedy użytkownicy wypełniają formularz, serwer musi wiedzieć, jakie informacje zostały podane w poszczególnych polach (np. w przypadku formularza logowania serwer musi wiedzieć, co zostało podane jako nazwa użytkownika, a co jako hasło). Właśnie z tego powodu w każdym polu formularza należy podać wartość atrybutu `name`. Identyfikuje ona pole formularza i jest przesyłana wraz z jego zawartością na serwer.

## maxlength

Atrybutu `maxlength` można używać w celu ograniczenia liczby znaków wpisywanych w polu. Jego wartość jest liczbą określającą, ile maksymalnie znaków można umieścić w danym polu. Jeśli np. oczekujemy od użytkownika podania roku, możemy umieścić w polu atrybut `maxlength` o wartości 4.

# POLE HASŁA

## <input>

### type="password"

Jeśli atrybutowi type zostanie przypisana wartość password, można utworzyć pole działające podobnie do zwykłego, jednowierszowego pola tekstowego z tą różnicą, że wpisywane do niego znaki będą maskowane. Dzięki temu nikt, zaglądając nam przez ramię, nie będzie mógł odczytać wpisywanych w polu wrażliwych danych, takich jak hasło.

### name

Atrybut name określa nazwę pola hasła, a jego wartość jest przesyłana na serwer wraz z informacjami podanymi przez użytkownika.

### size, maxlength

W polach haseł można także używać atrybutów size oraz maxlength, których znaczenie jest takie samo jak w przypadku jednowierszowych pól tekstowych.

rozdział-07/password-input.html

HTML

```
<form action="http://www.jakaswitryna.com/login.php">
  <p>Nazwa użytkownika:
    <input type="text" name="username" size="15"
      maxlength="30" />
  </p>
  <p>Hasło:
    <input type="password" name="password" size="15"
      maxlength="30" />
  </p>
</form>
```

WYNIK

Nazwa użytkownika:

Hasło:

Choć dane wpisywane w polu hasła są maskowane i nie można ich odczytać na ekranie, nie oznacza to wcale, że są one w bezpieczny sposób przesyłane na serwer. Nigdy nie należy używać tych pól do przesyłania wrażliwych informacji, takich jak numery kart kredytowych.

Aby zapewnić maksymalne bezpieczeństwo, należy skonfigurować serwer w taki sposób, by komunikował się z przeglądarką, używając protokołu Secure Socket Layer (SSL). Zagadnienia związane z SSL wykraczają poza zakres tematyczny tej książki, jednak na poświęconej jej stronie można znaleźć odnośniki do witryn zawierających więcej informacji na temat tego protokołu.

# WIELOWIERSZOWE POLA TEKSTOWE

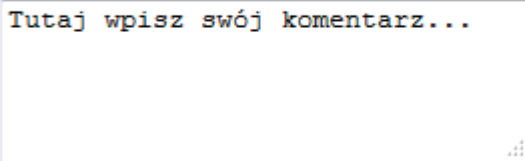
## HTML

rozdzial-07/textarea.html

```
<form action="http://www.jakaswitryna.com/login.php">
  <p>Co myślisz o tym koncercie?</p>
  <textarea name="comments" cols="30" rows="4">Tutaj
    wpisz swój komentarz...</textarea>
</form>
```

## WYNIK

Co myślisz o tym koncercie?



W przypadku tworzenia nowych formularzy szerokość oraz wysokość elementów `<textarea>` należy określać przy użyciu arkuszy stylów CSS. Niemniej podczas analizy kodu istniejących formularzy możemy się spotkać z zastosowaniem atrybutów `cols` oraz `rows`.

Atrybut `cols` określa szerokość pola (wyrażoną jako liczba znaków), natomiast atrybut `rows` — jego wysokość (wskazuje ona liczbę wierszy tekstu, które mają być widoczne w polu).

## `<textarea>`

Element `<textarea>` służy do tworzenia wielowierszowych pól tekstowych. W odróżnieniu od elementu `<input>` nie jest elementem pustym. Oznacza to, że powinien mieć zarówno znacznik otwierający, jak i zamykający.

Po wczytaniu strony cały tekst umieszczony pomiędzy otwierającym znacznikiem `<textarea>` oraz zamykającym znacznikiem `</textarea>` zostanie wyświetlony w polu.

Jeśli użytkownik nie usunie żadnych znaków umieszczonych początkowo w polu, ten sam komunikat zostanie przesłany na serwer wraz z dowolnym dopisanym do niego tekstem. (Na niektórych stronach używane są skrypty, które czyszczą zawartość tego pola, gdy użytkownik w nie kliknie).

# PRZYCISKI OPCJI

## <input>

### type="radio"

Przyciski opcji pozwalają użytkownikom wybrać tylko jedną opcję z grupy kilku opcji.

### name

Atrybut name jest przesyłany na serwer wraz z wartością opcji wybranej przez użytkownika. Jeśli odpowiedzi do pytania zadanego w formularzu są przedstawione w formie przycisków opcji, to wszystkie z nich muszą mieć taką samą wartość atrybutu name.

### value

Atrybut value określa wartość, jaka zostanie przesłana na serwer w razie zaznaczenia danego przycisku opcji. Wartości wszystkich przycisków opcji należących do jednej grupy powinny być unikalne (aby serwer wiedział, który przycisk opcji został zaznaczony przez użytkownika).

### checked

Atrybutu checked można użyć, by wskazać przycisk opcji, który powinien zostać zaznaczony w momencie wyświetlania strony. Atrybut ten powinien przyjmować wartość checked. Można go umieścić tylko w jednym przycisku opcji w danej grupie.

rozdzial-07/radio-button.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Wybierz swój ulubiony rodzaj muzyki:
  <br />
  <input type="radio" name="genre" value="rock"
    checked="checked" /> Rock
  <input type="radio" name="genre" value="pop" />
    Pop
  <input type="radio" name="genre" value="jazz" />
    Jazz
  </p>
</form>
```

WYNIK

Wybierz swój ulubiony rodzaj muzyki:

☒ Rock ☐ Pop ☐ Jazz

**Ważna uwaga:** kiedy użytkownik zaznaczy jeden z przycisków opcji, nie może już usunąć tego zaznaczenia. Może co najwyżej zaznaczyć inny przycisk. Jeśli zatem dajemy użytkownikowi do wyboru

tylko jedną opcję i chcemy, by można ją było zaznaczyć lub usunąć jej zaznaczenie (jak jest np. w przypadku pól służących do akceptacji warunków rejestracji), to powinniśmy skorzystać z pól wyboru.

# POLA WYBORU

## HTML

rozdzial-07/checkbox.html

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Wybierz swoją ulubioną usługę muzyczną:
  <br />
  <input type="checkbox" name="service"
    value="itunes" checked="checked" /> iTunes
  <input type="checkbox" name="service"
    value="lastfm" /> Last.fm
  <input type="checkbox" name="service"
    value="spotify" /> Spotify
  </p>
</form>
```

## WYNIK

Wybierz swoją ulubioną usługę muzyczną:

☒ iTunes ☐ Last.fm ☐ Spotify

## <input>

### type="checkbox"

Pola wyboru umożliwiają użytkownikom zaznaczanie (i usuwanie zaznaczenia) dowolnej liczby odpowiedzi na zadane pytanie.

### name

Atrybut name jest przesyłany na serwer wraz z wartościami pól zaznaczonych przez użytkownika. Jeśli odpowiedzi do konkretnego pytania zadanego w formularzu są przedstawione w formie pól wyboru, to wszystkie muszą mieć taką samą wartość atrybutu name.

### value

Atrybut value określa wartość, jaka zostanie przesłana na serwer w razie zaznaczenia danego pola wyboru.

### checked

Atrybutu checked można użyć, by wskazać pole wyboru, które powinno zostać zaznaczone w momencie wyświetlania strony. Atrybut ten powinien przyjmować wartość checked.

# LISTY ROZWIJANE

## <select>

Listy rozwijane pozwalają na wyświetlenie początkowo niewidocznej grupy opcji i wybranie jednej z nich.

Listy tego typu tworzy się przy użyciu elementu `<select>`. Wewnątrz niego umieszczane są dwa elementy `<option>` (lub więcej).

### name

Atrybut `name` określa nazwę pola formularza, która jest przesyłana na serwer wraz z wartością opcji wybranej przez użytkownika.

## <option>

Element `<option>` służy do tworzenia opcji listy, z których użytkownik może następnie wybierać. Słowa umieszczone pomiędzy otwierającym znacznikiem `<option>` a zamykającym znacznikiem `</option>` zostaną wyświetlone na liście.

### value

Element `<option>` korzysta z atrybutu `value`, by określić wartość, która, wraz z nazwą pola, zostanie przesłana na serwer po zaznaczeniu konkretnej opcji listy.

rozdzial-07/drop-down-list-box.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Jakiego urządzenia używasz do słuchania muzyki?</p>
  <select name="devices">
    <option value="ipod">iPoda</option>
    <option value="radio">radia</option>
    <option value="computer">komputera</option>
  </select>
</form>
```

WYNIK

Jakiego urządzenia używasz do słuchania muzyki?

iPoda	▼
iPoda	
radia	
komputera	

## selected

Atrybutu `selected` można użyć, by wskazać opcję, która ma być zaznaczona w momencie wyświetlenia strony. Atrybut ten powinien przyjmować wartość `selected`.

W razie pominięcia tego atrybutu w polu zostanie wyświetlona pierwsza podana opcja. Jeśli użytkownik nie wybierze innej opcji, to w momencie wysłania formularza na serwer zostanie przesłana wartość pierwszej opcji.

Lista rozwijana pełni podobną funkcję jak przyciski opcji (pozwala na wybór tylko jednej spośród kilku opcji). Oto dwa podstawowe czynniki decydujące o tym, które z tych pól należy zastosować:

1. Jeśli wszystkie dostępne opcje muszą być widoczne, to lepszym rozwiązaniem będą przyciski opcji.
2. Jeśli liczba dostępnych opcji jest bardzo duża (np. gdy mamy wybrać jeden z krajów świata), lepszym rozwiązaniem będzie zastosowanie listy rozwijanej.

# LISTA WIELOKROTNEGO WYBORU

## HTML

rozdzial-07/multiple-select-box.html

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Czy grasz na następujących instrumentach?
  (Możesz zaznaczyć więcej niż jedną opcję,
  naciskając klawisz Ctrl na komputerach PC
  lub klawisz Command na komputerach Mac).</p>
  <select name="instruments" size="3"
    multiple="multiple">
    <option value="guitar" selected="selected">
      gitara</option>
    <option value="drums">perkusja</option>
    <option value="keyboard"
      selected="selected">keyboard</option>
    <option value="bass">gitara basowa</option>
  </select>
</form>
```

## WYNIK

Czy grasz na następujących instrumentach? (Możesz zaznaczyć więcej niż jedną opcję, naciskając klawisz Ctrl na komputerach PC lub klawisz Command na komputerach Mac).



Dobrym rozwiązaniem jest poinformowanie użytkowników o możliwości zaznaczenia większej liczby opcji na liście. Warto także zaznaczyć, że na komputerach PC trzeba w tym celu wcisnąć klawisz *Ctrl* i klikać wybrane opcje, natomiast na komputerach Mac — wcisnąć klawisz *Command*.

## <select>

### size

Listę rozwijaną można zmienić w listę prezentującą jednocześnie więcej niż jedną opcję — wystarczy w elemencie `<select>` umieścić atrybut `size`. Wartością tego atrybutu jest liczba określająca, ile opcji ma być jednocześnie widocznych na liście. W przedstawionym przykładzie widać, że na liście widoczne są trzy spośród czterech opcji dostępnych na liście.

Niestety, implementacja tego atrybutu w różnych przeglądarkach nie jest idealna, dlatego w razie jego stosowania należy go dokładnie przetestować (zwłaszcza w przeglądarkach Firefox oraz Safari na komputerach Mac).

### multiple

Aby umożliwić użytkownikom wybór kilku spośród opcji dostępnych na liście, należy dodać do elementu `<select>` atrybut `multiple` o wartości `multiple`.

# POLE PRZESYŁANIA PLIKÓW

## <input>

Jeśli chcemy zapewnić użytkownikom możliwość przesyłania na serwer plików (takich jak obrazki, klipy wideo, pliki mp3 lub PDF), to konieczne będzie umieszczenie w formularzu odpowiedniego pola.

### type="file"

Zastosowanie tej wartości atrybutu type powoduje utworzenie pola przypominającego zwyczajnie jednowierszowe pole tekstowe oraz umieszczonego obok niego przycisku *Przełóżaj*. (Jedynym wyjątkiem jest przeglądarka Safari na komputerach Mac — zazwyczaj wyświetla ona przycisk *Wybierz plik*, za którym umieszczone jest pole tekstowe; początkowo, gdy żaden plik nie został wybrany, jest w nim wyświetlany tekst „Brak zaznaczonych plików”, natomiast gdy użytkownik wybierze plik, w polu zostanie wyświetlona jego nazwa). Kiedy użytkownik kliknie ten przycisk, na ekranie zostanie wyświetlone okno dialogowe pozwalające na wybranie jednego z plików dostępnych na komputerze.

rozdzial-07/file-input-box.html

HTML

```
<form action="http://www.jakaswitryna.com/upload.php"
  method="post">
  <p>Prześlij plik muzyczny w formacie MP3:</p>
  <input type="file" name="user-song" /><br />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Prześlij plik muzyczny w formacie MP3:

The screenshot shows a web form with the text "Prześlij plik muzyczny w formacie MP3:" above a file input field. To the right of the input field is a button labeled "Przełóżaj". Below the input field is a button labeled "Prześlij".

Jeśli formularz ma zapewniać możliwość przesyłania plików na serwer, to atrybut `method` elementu `<form>` musi mieć wartość `post`. (Plików nie można przesyłać przy użyciu metody `get` protokołu HTTP).

Kiedy użytkownik kliknie przycisk w polu przesyłania plików, wyświetlone okno dialogowe pozwalające na wybór pliku będzie miało wygląd dostosowany do używanego systemu operacyjnego. Nie ma możliwości zmiany wyglądu tego okna.



# PRZYCISK PRZESYŁAJĄCY

## HTML

rozdzial-07/submit-button.html

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Subskrybuj nasz biuletyn informacyjny:</p>
  <input type="text" name="email" />
  <input type="submit" name="subscribe"
    value="Subskrybuj" />
</form>
```

## WYNIK

Subskrybuj nasz biuletyn informacyjny:

## <input>

### type="submit"

Przyciski tego typu są używane do przesyłania formularzy na serwer.

### name

Można określić wartość atrybutu name, jednak nie ma ona żadnego znaczenia.

### value

Atrybut value służy do określania tekstu, który będzie wyświetlany na przycisku. Warto podać ten tekst, w przeciwnym razie na przycisku pojawi się komunikat „Wyślij zapytanie”, co w niektórych rodzajach formularzy może nie być właściwą nazwą przycisku.

W różnych przeglądarkach przycisk przesyłający może wyglądać nieco inaczej, jednak zazwyczaj jego wygląd odpowiada sposobowi prezentacji pól formularzy używanemu w danej przeglądarce. Jeśli zależy nam na określeniu wyglądu przycisków, możemy to zrobić za pomocą arkuszy stylów CSS (zgodnie z informacjami podanymi na stronie 342) lub użyć obrazka jako przycisku.

# PRZYCISKI Z OBRAZEM

## <image>

### type="image"

Jeśli chcemy, by przycisk przesyłający formularz miał postać obrazka, to atrybutowi type elementu <input> powinniśmy przypisać wartość image. W takim przypadku można użyć atrybutów: src, width, height oraz alt, działających dokładnie tak samo jak w elemencie <img> (opisanym na stronach 98 – 99).

rozdzial-07/image-button.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Subskrybuj nasz biuletyn informacyjny:</p>
  <input type="text" name="email" />
  <input type="image" src="images/subscribe.jpg"
    width="100" height="20" />
</form>
```

WYNIK

Subskrybuj nasz biuletyn informacyjny:

SUBSKRYBUJ

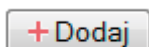
# ELEMENT BUTTON ORAZ POLA UKRYTE

## HTML

rozdzial-07/button-and-hidden-controls.html

```
<form action="http://www.jakaswitryna.com/profile.php">
  <button> Dodaj</button>
  <input type="hidden" name="bookmark"
    value="lirya" />
</form>
```

## WYNIK



## <button>

Element <button> został wprowadzony, by zapewnić twórcom stron większą kontrolę nad wyglądem przycisków oraz pozwolić na umieszczanie na nich innych elementów.

Oznacza to, że pomiędzy otwierającym znacznikiem <button> a zamykającym znacznikiem </button> można umieścić np. tekst i obraz.

## <input>

### type="hidden"

Ten przykład przedstawia zastosowanie ukrytych pól formularzy, tworzonych za pomocą atrybutu type o wartości hidden. Pola tego typu nie są wyświetlane na stronie (choć można je zobaczyć, wyświetlając jej kod źródłowy przy użyciu opcji *Pokaż źródło strony*). Pozwalają one autorom stron na dodawanie do formularzy wartości niewidocznych dla użytkowników. Autor witryny mógłby zastosować takie pole, by np. przekazać na serwer informację o tym, która strona była wyświetlona w momencie przesyłania formularza.

# DODAWANIE ETYKIET DO PÓL FORMULARZY

## <label>

Podczas prezentowania różnych rodzajów pól formularzy celowo zastosowano uproszczony kod, w którym tekst określający przeznaczenie każdego z pól był umieszczany bezpośrednio przy nim. Jednak każdemu z pól formularzy powinien towarzyszyć element `<label>`, który ułatwi korzystanie z formularzy osobom mającym problemy ze wzrokiem.

Element `<label>` może być używany na dwa sposoby:

1. Wewnątrz niego można umieścić zarówno opis tekstowy, jak i samo pole formularza (co widać w pierwszym wierszu przedstawionego przykładu).
2. Można go oddzielić od pola formularza i jednocześnie dodać do niego atrybut `for` określający, z którym polem skojarzona jest dana etykieta (jak zrobiono w przypadku przycisków opcji).

## for

Atrybut `for` określa, do którego pola formularza należy dana etykieta. Zwróćmy uwagę, że w przedstawionym przykładzie, w przyciskach opcji, zostały podane atrybuty `id`. Wartość tego atrybutu w jednoznaczny sposób identyfikuje element formularza, odróżniając go od wszystkich innych elementów strony (atrybut `id` został opisany na stronie 182).

rozdzial-07/labelling-form-controls.html

HTML

```
<label>Wiek: <input type="text" name="age" /></label>  
<br />
```

Płeć:

```
<input id="female" type="radio" name="gender"  
  value="f">  
<label for="female">kobieta</label>  
<input id="male" type="radio" name="gender"  
  value="m">  
<label for="male">mężczyzna</label>
```

WYNIK

Wiek:   
Płeć: ☐ kobieta ☐ mężczyzna

Wartość atrybutu `for` odpowiada wartości atrybutu `id` pola, do którego odnosi się dana etykieta. Tej techniki, bazującej na wykorzystaniu atrybutów `for` i `id`, można używać zawsze, niezależnie do typu pola formularza. W przypadku gdy element `<label>` jest używany wraz z polami wyboru lub przyciskami opcji, użytkownik może zaznaczyć dane pole, klikając zarówno je, jak i jego etykietę. Powiększenie obszaru, który użytkownik może kliknąć, ułatwia korzystanie z formularza. Bardzo duże znaczenie ma także położenie etykiety. Jeśli użytkownicy nie będą wiedzieć, gdzie należy wpisać informacje

lub jakie informacje wpisać w danym polu, zmniejsza się prawdopodobieństwo prawidłowego użycia formularza.

Oto zasady określające optymalne położenie etykiet względem pól formularzy.

## POWYŻEJ LUB Z LEWEJ STRONY:

- pola tekstowe,
- wielowierszowe pola tekstowe,
- listy rozwijane,
- pola przesyłania plików.

## Z PRAWEJ STRONY:

- poszczególne pola wyboru,
- poszczególne przyciski opcji.

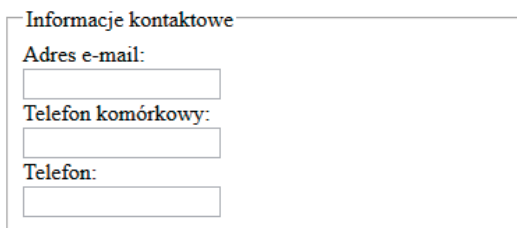
# GRUPOWANIE ELEMENTÓW FORMULARZY

HTML

rozdzial-07/grouping-form-elements.html

```
<fieldset>
  <legend>Informacje kontaktowe</legend>
  <label>Adres e-mail:<br />
  <input type="text" name="email" /></label><br />
  <label>Telefon komórkowy:<br />
  <input type="text" name="mobile" /></label><br />
  <label>Telefon:<br />
  <input type="text" name="telephone" /></label>
</fieldset>
```

WYNIK



## <fieldset>

Powiązane ze sobą pola formularza można zgrupować, umieszczając je wewnątrz elementu `<fieldset>`. Rozwiązanie to jest szczególnie wygodne w przypadku dłuższych formularzy.

W większości przeglądarek zawartość elementów `<fieldset>` jest wyświetlana wewnątrz obramowania, co wizualnie grupuje powiązane ze sobą elementy. Wygląd tego obramowania można określić przy użyciu arkusza stylów CSS.

## <legend>

Element `<legend>` można umieścić bezpośrednio za otwierającym znacznikiem `<fieldset>`. Zawiera on tekst, który ułatwi użytkownikowi zrozumienie przeznaczenia danej grupy pól formularza.

# HTML5: WERYFIKACJA PÓL FORMULARZY

Prawdopodobnie Czytelnik spotkał się już z formularzami, które prezentowały użytkownikom komunikaty w przypadku, gdy któreś z ich pól nie zostało prawidłowo wypełnione. Takie działanie jest nazywane **weryfikacją pól formularzy**.

Tradycyjnie weryfikacja pól formularzy była realizowana przy wykorzystaniu języka JavaScript (którego przedstawienie wykracza poza zakres tej książki). Jednak język HTML5 udostępnił mechanizmy takiej weryfikacji, przekazując związane z nią operacje przeglądarce.

Weryfikacja pól formularzy zapewnia, że po przesłaniu formularza na serwer informacje wpisane w polach zostaną prawidłowo zrozumiane. Weryfikacja treści przed przesłaniem formularza na serwer jest korzystna, gdyż:

- ogranicza czynności, które trzeba będzie wykonać na serwerze;
- sprawia, że użytkownik szybciej dowiaduje się o problemach z podanymi informacjami niż wtedy, gdy weryfikacja jest przeprowadzana na serwerze.

rozdzial-07/html5-form-validation.html

HTML

```
<form action="http://www.jakaswitryna.com/login.php"
  method="post">
  <label for="username">Nazwa użytkownika:</label>
  <input type="text" name="username"
    required="required" /></title><br />
  <label for="password">Hasło:</label>
  <input type="password" name="password"
    required="required" />
  <input type="submit" value="Zaloguj się" />
</form>
```

WYNIK

Nazwa użytkownika: Ewa

Hasło:

Zaloguj się

Proszę wypełnić to pole.

Gdy powstawała ta książka, jedynie przeglądarki Chrome i Opera obsługiwały weryfikację pól formularzy w HTML5, choć należy się spodziewać, że inne przeglądarki pójdą w ich ślady. By zapewnić weryfikację formularzy także w starszych przeglądarkach (które nie obsługują języka HTML5), twórcy stron będą zapewne wciąż używać języka JavaScript.

W tym przykładzie weryfikacja bazuje na zastosowaniu atrybutu `required` — można go umieścić w dowolnym polu formularza, które użytkownik powinien wypełnić. Ten atrybut języka HTML5 nie wymaga określania wartości, co odróżnia go od atrybutów w języku HTML 4, w którym określanie wartości atrybutów było konieczne. Niektórzy twórcy stron przypisują temu atrybutowi wartość `required`.

# HTML5: POLE DATY

## HTML

rozdzial-07/html5-date-input.html

```
<form action="http://www.jakaswitryna.com/bookings"
  method="post">
  <label for="username">Data wylotu:</label>
  <input type="date" name="depart" />
  <input type="submit" value="Prześlij" />
</form>
```

## WYNIK

Data wylotu:

## <input>

Wiele formularzy musi pobierać takie informacje jak daty, adresy e-mail oraz adresy URL. Tradycyjnie do tego celu używane były zwyczajne pola tekstowe.

Język HTML5 jednak wprowadza nowe pola, które mają służyć standaryzacji sposobów wprowadzania niektórych informacji. Starsze przeglądarki, które nie rozpoznają tych typów pól, będą je wyświetlać jako zwyczajne, jednowierszowe pola tekstowe.

## type="date"

Jeśli chcemy prosić użytkownika o podanie daty, możemy w tym celu użyć elementu <input>, w którym atrybutowi type przypisana zostanie wartość date. Dzięki temu w przeglądarkach obsługujących HTML5 wyświetlone zostanie specjalne pole służące do wyboru daty.

Ten przykład pokazuje, jak wygląda pole wyboru daty w przeglądarce Opera. W innych przeglądarkach pole to może wyglądać inaczej.

# HTML5: POLA ADRESÓW E-MAIL I URL

## <input>

HTML5 udostępnia pola pozwalające użytkownikom na podawanie adresów poczty elektronicznej oraz adresów URL. Przeglądarki, które nie obsługują tych typów, będą wyświetlały te pola jako zwykajne, jednowierszowe pola tekstowe.

### type="email"

Jeśli prosimy użytkownika o podanie adresu e-mail, możemy skorzystać z nowego typu pola tekstowego. Dzięki temu przeglądarki obsługujące weryfikację danych w HTML5 będą sprawdzać, czy podane informacje zostały zapisane w prawidłowym formacie, charakterystycznym dla adresów poczty elektronicznej. Niektóre smartfony optymalizują prezentowaną klawiaturę, wyświetlając na niej przyciski, które najprawdopodobniej będą potrzebne do wpisania adresu e-mail (np. przycisk z symbolem @).

### type="url"

Jeśli użytkownik ma podać adres strony WWW, można skorzystać z pola adresu URL. Dzięki temu przeglądarki obsługujące weryfikację danych w HTML5 będą mogły sprawdzić, czy użytkownik wpisał informacje w formacie

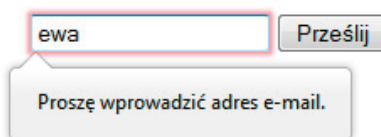
rozdzial-07/html5-email-input.html

HTML

```
<form action="http://www.jakaswitryna.com/subscribe.php">
  <p>Proszę podać swój adres e-mail:</p>
  <input type="email" name="email" />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Proszę podać swój adres e-mail:



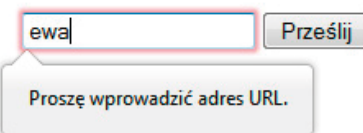
rozdzial-07/html5-url-input.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Proszę podać adres swojej witryny:</p>
  <input type="url" name="website" />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Proszę podać adres swojej witryny:



charakterystycznym dla adresów URL. Niektóre smartfony optymalizują

klawiaturę, wyświetlając na niej przyciski przydatne do wpisywania adresów.



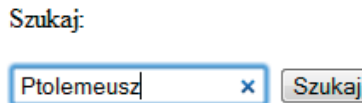
# HTML5: POLE WYSZUKIWANIA

## HTML

rozdzial-07/html5-search-input.html

```
<form action="http://www.jakaswitryna.com/search.php">
  <p>Szukaj:</p>
  <input type="search" name="search" />
  <input type="submit" value="Szukaj" />
</form>
```

## WYNIK

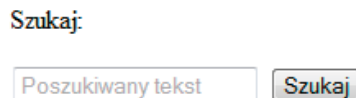


## HTML

rozdzial-07/html5-placeholder.html

```
<form action="http://www.jakaswitryna.com/search.php">
  <p>Szukaj:</p>
  <input type="search" name="search"
    placeholder="Poszukiwany tekst"/>
  <input type="submit" value="Szukaj" />
</form>
```

## WYNIK



## placeholder

W dowolnym polu tekstowym można także użyć atrybutu o nazwie placeholder, którego wartość określa tekst, jaki będzie prezentowany w polu do momentu, gdy użytkownik je kliknie. Starsze przeglądarki ignorują ten atrybut.

## <input>

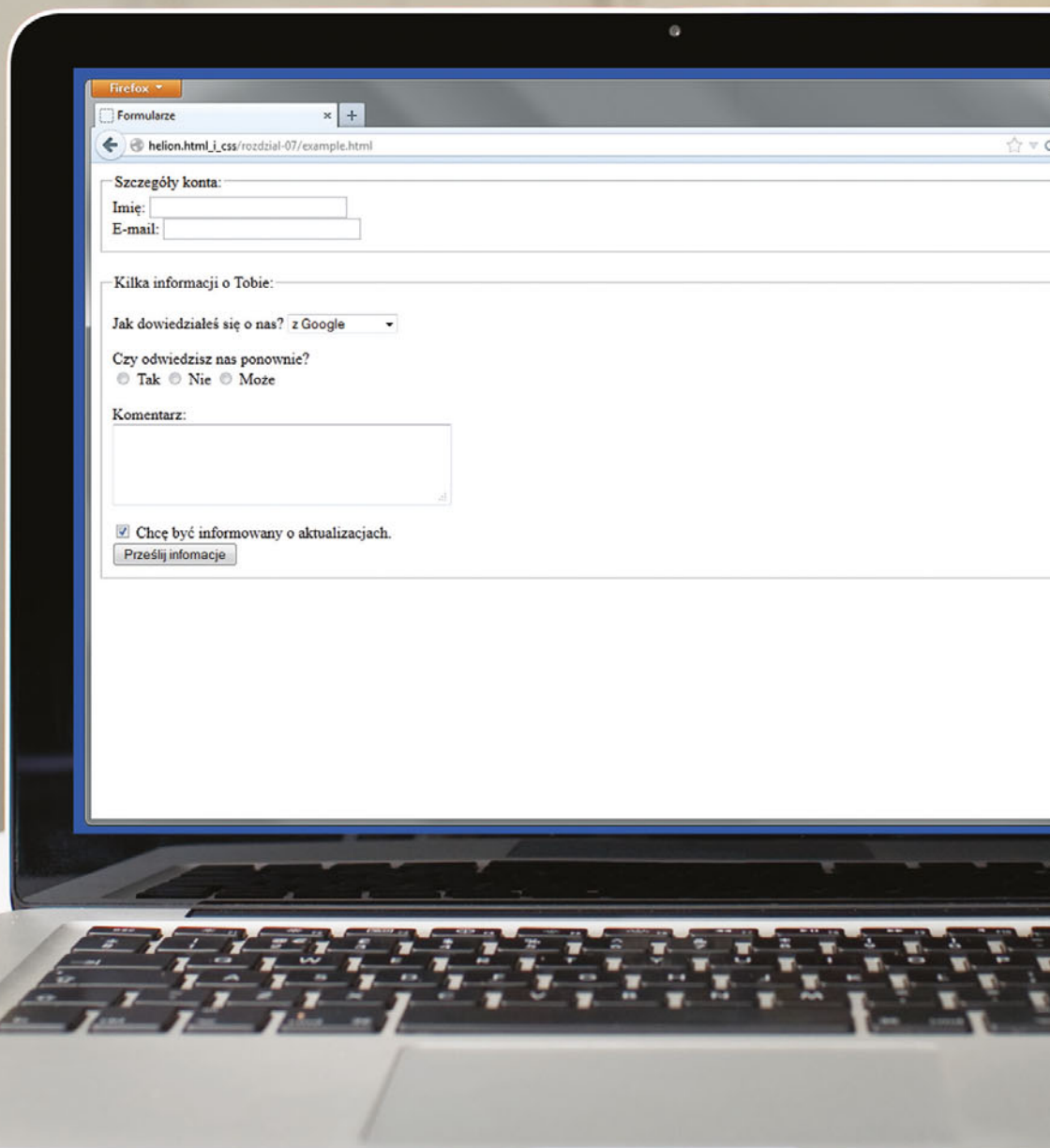
Jeśli chcemy umieścić na stronie pojedyncze pole tekstowe służące do wyszukiwania, HTML5 ułatwia nam to, udostępniając specjalny typ pola przeznaczonego właśnie do tego celu.

## type="search"

Jeśli chcemy umieścić na stronie pojedyncze pole tekstowe służące do wyszukiwania, HTML5 ułatwia nam to, udostępniając specjalne pole wyszukiwania.

Aby utworzyć takie pole, należy do elementu <input> dodać atrybut type o wartości search. Starsze przeglądarki będą je traktować jako zwyczajne, jednowierszowe pole tekstowe.

Najnowsze przeglądarki rozszerzają nieco możliwości pól tego typu, poprawiając ich użyteczność. I tak w przeglądarce Safari na komputery Mac zostało dodane niewielkie kółko z krzyżykiem pozwalające wyczyścić zawartość pola, kiedy chcemy wpisać w nim zupełnie nowe słowo. Dodatkowo przeglądarka ta automatycznie zaokrągla wierzchołki tego pola.

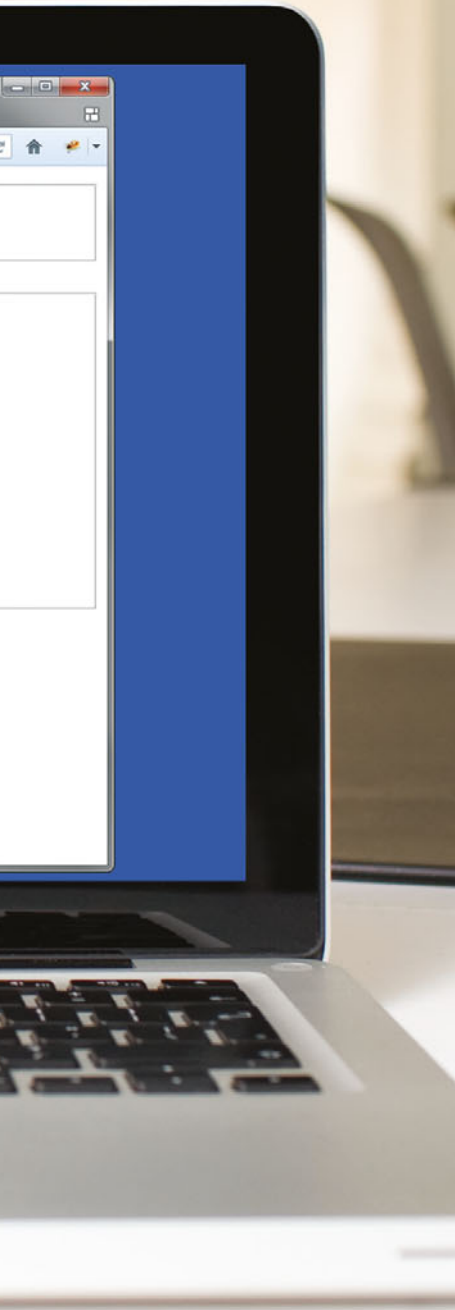


# PRZYKŁAD

## FORMULARZE

Ten przykład przedstawia formularze do przesyłania komentarzy oraz subskrypcji biuletynu informacyjnego. Wykorzystano w nich kilka typów pól.

Element `<form>` korzysta z atrybutu `action`, by określić adres strony, do której zostaną przesłane dane. Każdy element reprezentujący pole formularza jest umieszczony wewnątrz elementu `<form>`. Różne typy pól formularzy są dostosowane do pobierania różnych typów danych. Element `<fieldset>` służy do grupowania pól powiązanych ze sobą. Element `<label>` określa przeznaczenie każdego z pól formularza.



# PRZYKŁAD

## FORMULARZE

```
<html>
  <head>
    <title>Formularze</title>
  </head>
  <body>
    <form action="http://www.jakaswitryna.com/review.php" method="get">
      <fieldset>
        <legend>
          Szczegóły konta:
        </legend>
        <label>
          Imię:
          <input type="text" name="name" size="30" maxlength="100">
        </label>
        <br />
        <label>
          E-mail:
          <input type="email" name="email" size="30" maxlength="100">
        </label>
        <br />
      </fieldset>
      <br />
      <fieldset>
        <legend>
          Kilka informacji o Tobie:
        </legend>
        <p>
          <label for="hear-about">
            Jak się o nas dowiedziałeś?
          </label>
          <select name="referrer" id="hear-about">
            <option value="google">z Google</option>
            <option value="friend">od znajomego</option>
            <option value="advert">z reklamy</option>
            <option value="other">w inny sposób</option>
          </select>
        </p>
      </fieldset>
    </form>
  </body>
</html>
```

# PRZYKŁAD

## FORMULARZE

```
Czy odwiedziś nas ponownie?  
<br />  
<label>  
  <input type="radio" name="rating" value="yes" />  
  Tak  
</label>  
<label>  
  <input type="radio" name="rating" value="no" />  
  Nie  
</label>  
<label>  
  <input type="radio" name="rating" value="maybe" />  
  Może  
</label>  
</p>  
<p>  
  <label for="comments">  
    Komentarz:  
  </label>  
  <br />  
  <textarea rows="4" cols="40" id="comments">  
  </textarea>  
</p>  
<label>  
  <input type="checkbox" name="subscribe" checked="checked" />  
  Chcę być informowany o aktualizacjach.  
</label>  
<br />  
  <input type="submit" value="Prześlij informacje" />  
</fieldset>  
</form>  
</body>  
</html>
```

# PODSUMOWANIE

## FORMULARZE

- ▶ Zawsze, gdy chcemy odbierać informacje od użytkowników, musimy skorzystać z formularza umieszczonego wewnątrz znacznika `<form>`.
- ▶ Informacje podawane w formularzach są przesyłane w postaci par nazwa – wartość.
- ▶ Każde pole formularza musi mieć swoją nazwę; wpisywany w nim tekst lub wybierana w nim wartość są przesyłane na serwer.
- ▶ Język HTML5 udostępnia nowe typy pól formularzy, ułatwiające użytkownikom wprowadzanie danych.

# 14

## LISTY, TABELE I FORMULARZE

- ▶ Określanie postaci znaczników list.
- ▶ Dodawanie krawędzi oraz tła do tabel.
- ▶ Zmiana wyglądu elementów formularzy.

# OKREŚLANIE POSTACI FORMULARZY

Chyba nikt nie lubi wypełniania formularzy, dlatego jeśli postaramy się, by formularze na stronach były bardziej atrakcyjne i łatwiejsze w użyciu, więcej użytkowników je wypełni. Co więcej, jeśli spróbujemy wyświetlić formularz w kilku różnych przeglądarkach (co pokazują rysunki zamieszczone na tej stronie), okaże się, że każda z nich wyświetla go w nieco inny sposób.

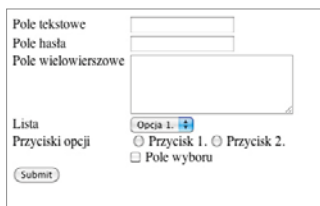
Kaskadowe arkusze stylów są bardzo często używane do określania postaci elementów formularzy. Ma to nie tylko poprawić ich atrakcyjność, lecz także sprawić, by ich wygląd w różnych przeglądarkach był bardziej spójny.

Najczęściej określany jest wygląd:

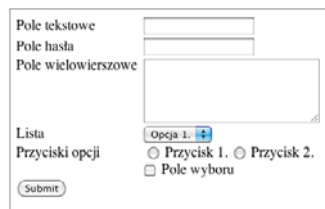
- pól tekstowych oraz wielowierszowych pól tekstowych,
- przycisków przesyłających formularze,
- etykiet — w celu ich efektownego wyrównania.

Na kolejnych stronach książki będzie mowa o tym, jak realizować te zadania przy wykorzystaniu CSS.

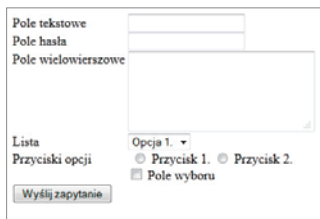
## MAC: SAFARI



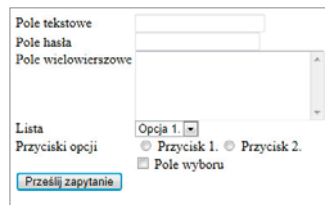
## PC: FIREFOX



## MAC: CHROME



## PC: INTERNET EXPLORER



Określanie postaci pól tekstowych oraz przycisków jest stosunkowo proste. Bardziej kłopotliwe jest tworzenie stylów, które zapewnią spójny wygląd list rozwijanych, przycisków opcji oraz pól wyboru w różnych przeglądarkach.

Aby uzyskać taki efekt, warto pobrać arkusz CSS dostępny na stronie <http://formalize.me>.

Autor tej strony włożył bardzo wiele pracy, by zapewnić spójny wygląd formularzy w wielu przeglądarkach. Choć zastosowane rozwiązanie korzysta z języka JavaScript, to aby móc z niego skorzystać, nie trzeba się znać na programowaniu.



# OKREŚLANIE POSTACI PÓL TEKSTOWYCH

## CSS

rozdzial-14/styling-text-inputs.html

```
input {
  font-size: 120%;
  color: #5a5854;
  background-color: #f2f2f2;
  border: 1px solid #bdbdbd;
  border-radius: 5px;
  padding: 5px 5px 5px 30px;
  background-repeat: no-repeat;
  background-position: 8px 9px;
  display: block;
  margin-bottom: 10px;}
input:focus {
  background-color: #ffffff;
  border: 1px solid #b1e1e4;}
input#email {
  background-image: url("images/email.png");}
input#twitter {
  background-image: url("images/twitter.png");}
input#web {
  background-image: url("images/web.png");}
```

## WYNIK



Więcej informacji na temat obrazów tła oraz sposobów ich umiejscawiania można znaleźć w rozdziale 16.

Zamieszczony tu przykład przedstawia właściwości CSS powszechnie stosowane do określania postaci pól tekstowych. Większość z nich została już omówiona we wcześniejszych rozdziałach książki.

Właściwość **font-size** określa wielkość czcionki używanej do wpisywania tekstu w polach tekstowych.

Właściwość **color** określa kolor tekstu, a **background-color** — kolor tła pól tekstowych.

Właściwość **border** pozwala wyświetlić obramowanie pola, a **border-radius** umożliwia stworzenie zaokrąglonych wierzchołków (w przeglądarkach, które na to pozwalają).

Pseudoklasa **:focus** służy do zmiany koloru tła pola, w którym aktualnie użytkownik coś wpisuje, a pseudoklasa **:hover** pozwala zastosować te same style w elementach, które użytkownik wskazał myszką.

Właściwość **background-image** określa obraz tła używany w polach. Ponieważ w każdym z pól został zastosowany inny obraz, skorzystaliśmy z selektora identyfikatora, by określić postać konkretnych elementów na podstawie wartości ich atrybutów id.

# OKREŚLANIE POSTACI PRZYCISKÓW PRZESYŁAJĄCYCH

Na tej stronie przedstawione zostały niektóre właściwości, których można użyć do określania postaci przycisków przesyłających. Prezentowany przykład bazuje na arkuszu zamieszczonym na poprzedniej stronie, a przyciski przesyłające dziedziczą style określone dla elementu `<input>`.

Właściwość `color` służy do zmiany koloru tekstu wyświetlanego na przyciskach.

Właściwość `text-shadow` pozwoli zapewnić trójwymiarowy wygląd tekstu, o ile tylko używana przeglądarka ją obsługuje.

Właściwość `border-bottom` została użyta do pogrubienia dolnego obramowania przycisku, dzięki czemu wygląda on nieco bardziej przestrzennie.

Właściwość `background-color` pozwala wyróżnić przycisk spośród innych otaczających go elementów. (Stworzenie jednego, spójnego sposobu prezentacji przycisków ułatwi użytkownikom zrozumienie, jak należy korzystać z witryny). Do przycisku dodano także tło o postaci gradientu, które zostanie wykorzystane, jeśli tylko przeglądarka zapewnia taką możliwość. Zagadnienia związane z tworzeniem i stosowaniem gradientów zostały omówione na stronie 418.

rozdzial-14/styling-submit-buttons.html

CSS

```
input#submit {
  color: #444444;
  text-shadow: 0px 1px 1px #ffffff;
  border-bottom: 2px solid #b2b2b2;
  background-color: #b9e4e3;
  background: -webkit-gradient(linear, left top,
    left bottom, from(#beee9), to(#a8cfce));
  background:
    -moz-linear-gradient(top, #beee9, #a8cfce);
  background:
    -o-linear-gradient(top, #beee9, #a8cfce);
  background:
    -ms-linear-gradient(top, #beee9, #a8cfce);}
input#submit:hover {
  color: #333333;
  border: 1px solid #a4a4a4;
  border-top: 2px solid #b2b2b2;
  background-color: #a0dbc4;
  background: -webkit-gradient(linear, left top,
    left bottom, from(#a8cfce), to(#beee9));
  background:
    -moz-linear-gradient(top, #a8cfce, #beee9);
  background:
    -o-linear-gradient(top, #a8cfce, #beee9);
  background:
    -ms-linear-gradient(top, #a8cfce, #beee9);}
```

Rejestruj

WYNIK

Do zmiany wyglądu przycisku po wskazaniu go myszą została zastosowana pseudoklasa `:hover`. W tym przypadku umieszczenie wskaźnika myszy na przycisku spowoduje zmianę koloru jego tła oraz zmianę koloru tekstu (na nieco ciemniejszy), a także pogrubienie dolnego obramowania przycisku.

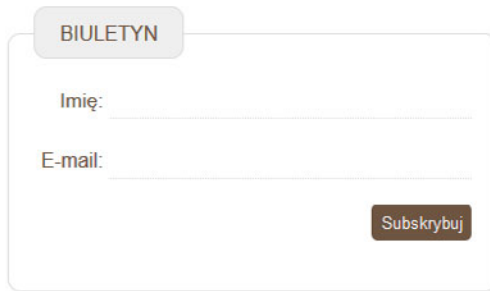
# OKREŚLANIE POSTACI GRUP PÓL ORAZ ICH OPISU

## CSS

rozdzial-14/styling-fieldsets-and-legends.html

```
fieldset {
  width: 350px;
  border: 1px solid #dcdcdc;
  border-radius: 10px;
  padding: 20px;
  text-align: right;}
legend {
  background-color: #efefef;
  border: 1px solid #dcdcdc;
  border-radius: 10px;
  padding: 10px 20px;
  text-align: left;
  text-transform: uppercase;}
```

## WYNIK



Grupy pól — elementy `<fieldset>` — są szczególnie przydatne podczas określania obszaru formularzy.

W przypadku bardzo dużych formularzy można ich używać do grupowania pól, które są ze sobą powiązane.

Z kolei ich opis — element `<legend>` — określa, jakie informacje mają być podane w danej grupie pól.

Podczas określania postaci tych dwóch elementów powszechnie stosowane są następujące właściwości.

Właściwość `width`, której używa się do określania szerokości grupy pól.

W zamieszczonym przykładzie narzucona szerokość wymusza, by pola zostały wyświetlone w dwóch wierszach. (Gdyby szerokość była większa, to pola mogłyby zostać wyświetlone w jednym wierszu).

Właściwość `color` służy do określenia koloru tekstu.

Kolor tła grupy pól oraz jej opisu jest określany przy użyciu właściwości `background-color`.

Właściwość `border` określa postać obramowania wyświetlanego wokół grupy pól oraz jej opisu.

Właściwość `border-radius` pozwala zaokrąglić wierzchołki obramowania elementów w przeglądarkach, które ją obsługują.

I wreszcie właściwość `padding` została użyta, by nieco zwiększyć wypełnienie wewnątrz tych elementów.

# WYRÓWNYWANIE ELEMENTÓW FORMULARZY: PREZENTACJA PROBLEMU

Zazwyczaj etykiety poszczególnych pól formularzy mają różne długości, co oznacza, że pola nie będą wyświetlane równo jedno pod drugim. Taką sytuację wyraźnie widać na przykładzie przedstawionym u dołu strony (w którym nie są używane żadne style CSS).

W tym przykładowym formularzu każde zagadnienie zostało umieszczone w odrębnym elemencie `<div>`, aby poszczególne pytania i związane z nim pola były prezentowane w osobnym wierszu. Niemniej łatwiej jest wypełniać formularze, jeśli ich pola są ładnie wyrównane jedno pod drugim. Problem ten rozwiązuje arkusz stylów przedstawiony na następnej stronie.

W wierszu, w którym użytkownik jest proszony o określenie płci, zostały umieszczone dwa elementy `<label>` (zawierające odpowiednio słowa: mężczyzna, kobieta). Sam tytuł tego wiersza został umieszczony w elemencie `<span>`, dzięki czemu łatwiej go będzie wyrównać.

Na poprzedniej stronie przedstawiona została technika wyrównywania elementów formularzy. Jeśli formularz zawiera wyłącznie pola

rozdzial-14/aligning-form-controls-problem.html

HTML

```
<form action="example.php" method="post">
  <div>
    <label for="name" class="title">Imię:</label>
    <input type="text" id="name" name="name" />
  </div>
  <div>
    <label for="email" class="title">E-mail:</label>
    <input type="email" id="email" name="email" />
  </div>
  <div>
    <span class="title">Płeć:</span>
    <input type="radio" name="gender" id="male"
      value="M" />
    <label for="male">M</label>
    <input type="radio" name="gender" id="female"
      value="F" />
    <label for="female">K</label><br />
  </div>
  <div>
    <input type="submit" value="Rejestruj"
      id="submit" />
  </div>
</form>
```

WYNIK BEZ UŻYCIA CSS

Imię:   
E-mail:   
Płeć: ☐ M ☐ K

tekstowe, wystarczy nadać im taką samą szerokość i wyrównać wszystkie elementy do prawej — dzięki temu pola formularza zostaną rozmieszczone idealnie jedno pod drugim, podobnie jak

etykiety. Jednak w przypadku bardziej złożonych formularzy konieczne będzie zastosowanie rozwiązania przypominającego to przedstawione tutaj.

# WYRÓWNYWANIE ELEMENTÓW FORMULARZY: ROZWIĄZANIE PROBLEMU

## CSS

rozdzial-14/aligning-form-controls-solution.html

```
div {  
  border-bottom: 1px solid #efefef;  
  margin: 10px;  
  padding-bottom: 10px;  
  width: 260px;}  
.title {  
  float: left;  
  width: 100px;  
  text-align: right;  
  padding-right: 10px;}  
.radio-buttons label {  
  float: none;}  
.submit {  
  text-align: right;}
```

## WYNIK

The screenshot shows a web form with three rows. Each row has a label on the left and an input field on the right. The labels are 'Imię:', 'E-mail:', and 'Płeć:'. The input fields are text boxes for the first two and radio buttons for the last one. The labels are right-aligned, and the input fields are left-aligned, creating a clean, aligned look. A 'Rejestruj' button is located at the bottom right of the form.

Odrębna reguła stylów określa postać elementów `<div>`, wewnątrz których umieszczone zostały poszczególne wiersze formularza (konkretnie precyzują one szerokość tych wierszy oraz odstęp pomiędzy nimi). W końcu przycisk przesyłający formularz został wyrównany do prawej.

Każdy wiersz formularza zawiera tytuł informujący użytkowników, co mają w nim wpisywać. W przypadku pól tekstowych takie tytuły są tworzone przy użyciu elementu `<label>`. Natomiast w wierszu zawierającym przyciski opcji został w tym celu zastosowany element `<span>`. W obu przypadkach w tych elementach został umieszczony atrybut `class` o wartości `title`.

Aby przesunąć tytuły pól na lewą stronę, możemy skorzystać z właściwości o nazwie `float`. (Została ona bardziej szczegółowo opisana na stronach 369 – 375).

Dzięki określeniu szerokości elementów (przy użyciu właściwości `width`) wiemy, że tytuły będą miały taką samą szerokość. Dlatego wyświetlone obok nich pola formularza zostaną idealnie wyrównane.

Po użyciu właściwości `text-align` tytuły pól zostały wyrównane do prawej, natomiast użycie właściwości `padding` sprawiło, że pomiędzy tekstem tytułu a polami formularza jest pewien odstęp.

# STYLE WSKAŹNIKA MYSZY

## cursor

Właściwość cursor pozwala określać postać wskaźnika myszy wyświetlanego przez przeglądarkę.

Możemy się np. zdecydować na to, by wskaźnik wyświetlany w obszarze formularza miał postać dłoni z wyciągniętym palcem wskazującym.

Poniżej przedstawiono najczęściej używane wartości właściwości cursor:

auto  
corsshair  
default  
pointer  
move  
text  
wait  
help  
url("cursor.gif");

Powyższych wartości należy używać wyłącznie w celu dodawania przydatnych dla użytkowników informacji w tych miejscach strony, w których mogą się oni spodziewać odpowiedniego wskaźnika myszy (np. wyświetlanie wskaźnika o postaci czterokierunkowej strzałki po wskazaniu łącza może zmylić użytkowników).

rozdzial-14/cursor.html

HTML

```
<a href="http://pl.wikipedia.org">Wikipedia</a>
```

```
a {  
  cursor: move;}
```

CSS

WYNIK

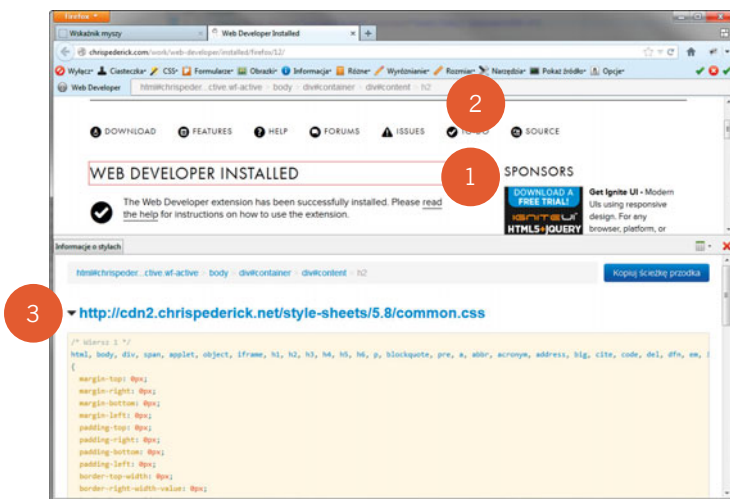


# WEB DEVELOPER TOOLBAR

To bardzo przydatne rozszerzenie do przeglądarek Firefox i Chrome udostępnia narzędzia pozwalające wyświetlać style CSS, które odnoszą się do elementu wskazanego myszką, oraz strukturę kodu HTML.

Opisywane tu narzędzie można pobrać ze strony <http://chrispederick.com/work/web-developer/>.

Aby wyświetlić style CSS oraz strukturę kodu HTML strony, należy otworzyć menu CSS narzędzia i wybrać opcję *View Style Information*.



## 1. ZARYSY ELEMENTÓW

Po wskazaniu elementu myszką wokół niego wyświetlane jest czerwone obramowanie, które doskonale pokazuje, jaki obszar jest zajmowany przez dany element.

## 2. STRUKTURA

Po wskazaniu elementu myszką u góry okna przeglądarki zostanie wyświetlona struktura kodu HTML. W tym przykładzie widać, że element `<li>` został przypisany do klasy `completed` i znajduje się wewnątrz elementu `<ul>` przypisanego do klasy `to-do`. Lista natomiast znajduje się wewnątrz elementu `<div>` o identyfikatorze `page`, umieszczonego w elementach `<body>` oraz `<html>`.

Narzędzie to może być bardzo pomocne podczas tworzenia reguł CSS, gdyż pomaga sprawdzić, czy odnoszą się one do odpowiednich elementów.

## 3. STYLE CSS

Po umieszczeniu wskaźnika myszy na elemencie można kliknąć, by wyświetlić odnoszące się do niego reguły stylów (wraz z informacjami o numerach wierszy, w których zostały podane). Powyżej reguł wyświetlana jest nazwa pliku arkusza stylów (oraz ścieżka dostępu do niego).

Rozwiązanie to znacznie upraszcza określanie, które reguły stylów odnoszą się do elementu. Można go używać zarówno podczas testowania własnej witryny, jak i do analizy stylów używanych na innych stronach.

Narzędzie to pozwala także na zmianę wielkości ekranu, weryfikację poprawności kodu HTML i CSS oraz wyłaczanie obrazków.

## Warsztaty poetyckie

W tym roku mamy zamiar zorganizować i przeprowadzić serię warsztatów oraz sympozjów poetyckich.

Dla naszych członków wstęp na te imprezy będzie bezpłatny.

- Perspektywa poetycka
- Wojna w oczach Baczyńskiego
- Poematy odnalezione i poeci-outsiderzy

	Kraków	Wrocław	Gdańsk
Perspektywa poetycka	Sobota, 4 lutego 2012 11:00 - 14:00	Sobota, 3 marca 2012 11:00 - 14:00	Sobota, 17 marca 2012 11:00 - 14:00
Wojna w oczach Baczyńskiego	Sobota, 7 kwietnia 2012 11:00 - 13:00	Sobota, 5 maja 2012 11:00 - 13:00	Sobota, 19 maja 2012 11:00 - 13:00
Poematy odnalezione i poeci-outsiderzy	Sobota, 9 czerwca 2012 11:00 - 14:00	Sobota, 7 lipca 2012 11:00 - 14:00	Sobota, 21 lipca 2012 11:00 - 14:00
Naturalna śmierć: Granice poznania	Sobota, 4 sierpnia 2012 11:00 - 16:00	Sobota, 8 września 2012 11:00 - 16:00	Sobota, 15 września 2012 11:00 - 16:00

### Rejestracja uczestnictwa

Imię i nazwisko:

E-mail:

Najbliższa placówka:

Czy jesteś członkiem stowarzyszenia?

☐ Tak ☐ Nie

Rejestracja





# PRZYKŁAD

## LISTY, TABELE I FORMULARZE

Zamieszczony tu przykład przedstawia zastosowanie kilku poznanych w tym rozdziale właściwości służących do określania wyglądu list, tabel oraz formularzy.

Znaczniki listy spotkań poetyckich, umieszczonej na początku strony, zostały wyświetlone przy użyciu obrazów. Odpowiednie odstępy pomiędzy poszczególnymi elementami listy zapewnia właściwość `line-height`.

Jeśli natomiast chodzi o umieszczoną poniżej tabelę, odstępy pomiędzy komórkami usunięto dzięki użyciu właściwości `border-spacing`. Określona została także wielkość czcionki używanej w elementach `<td>` oraz `<th>`, gdyż nie jest ona dziedziczona po przodkach.

Nagłówek tabeli ma ciemniejsze tło, jaśniejszy kolor tekstu i jest oddzielony do pozostałej zawartości tabeli linią o szerokości 2 pikseli. Okrągłe wierzchołki w nagłówku tabeli zostały utworzone przy użyciu pseudoklas: `:first-child` oraz `:last-child` (będą one widoczne w przeglądarkach, które na to pozwalają).

Tło w wierszach tabeli jest naprzemiennie zmieniane, a dodatkowo została zastosowana tekstura z różnymi obramowaniami po obu stronach komórek.

W przedstawionym formularzu powiązane ze sobą pola zostały umieszczone wewnątrz elementu `<fieldset>`. W etykietach wyświetlonych z lewej strony została użyta właściwość `float`, aby zapewnić, że pola formularza będą wyrównane w pionie.

Kiedy użytkownik kliknie pole tekstowe formularza albo umieści na nim wskaźnik myszy, zmieni się kolor tła oraz kolor obramowań pola. Zmodyfikowany zostanie także wygląd przycisku przesyłającego, by użytkownicy nie mieli wątpliwości, w jaki sposób należy przekazać dane na serwer.

# PRZYKŁAD

## LISTY, TABELE I FORMULARZE

```
<!DOCTYPE html>
<html>
  <head>
    <title>Listy, tabele i formularze</title>
    <style type="text/css">
      body {
        font-family: Arial, Verdana, sans-serif;
        font-size: 90%;
        color: #666;
        background-color: #f8f8f8;}
      li {
        list-style-image: url("images/icon-plus.png");
        line-height: 1.6em;}
      table {
        border-spacing: 0px;}
      th, td {
        padding: 5px 30px 5px 10px;
        border-spacing: 0px;
        font-size: 90%;
        margin: 0px;}
      th, td {
        text-align: left;
        background-color: #e0e9f0;
        border-top: 1px solid #f1f8fe;
        border-bottom: 1px solid #cbd2d8;
        border-right: 1px solid #cbd2d8;}
      tr.head th {
        color: #fff;
        background-color: #90b4d6;
        border-bottom: 2px solid #547ca0;
        border-right: 1px solid #749abe;
        border-top: 1px solid #90b4d6;
        text-align: center;
        text-shadow: -1px -1px 1px #666666;
        letter-spacing: 0.15em;}
      td {
        text-shadow: 1px 1px 1px #ffffff;}
```

# PRZYKŁAD

## LISTY, TABELE I FORMULARZE

```
tr.even td, tr.even th {
    background-color: #e8eff5;}
tr.head th:first-child {
    -webkit-border-top-left-radius: 5px;
    -moz-border-radius-topleft: 5px;
    border-top-left-radius: 5px;}
tr.head th:last-child {
    -webkit-border-top-right-radius: 5px;
    -moz-border-radius-topright: 5px;
    border-top-right-radius: 5px;}
fieldset {
    width: 310px;
    margin-top: 20px;
    border: 1px solid #d6d6d6;
    background-color: #ffffff;
    line-height: 1.6em;}
legend {
    font-style: italic;
    color: #666666;}
input[type="text"] {
    width: 120px;
    border: 1px solid #d6d6d6;
    padding: 2px;
    outline: none;}
input[type="text"]:focus,
input[type="text"]:hover {
    background-color: #d0e2f0;
    border: 1px solid #999999;}
input[type="submit"] {
    border: 1px solid #006633;
    background-color: #009966;
    color: #ffffff;
    border-radius: 5px;
    padding: 5px;
    margin-top: 10px;}
input[type="submit"]:hover {
    border: 1px solid #006633;
```

# PRZYKŁAD

## LISTY, TABELE I FORMULARZE

```
background-color: #00CC33;
color: #ffffff;
cursor: pointer;}
.title {
float: left;
width: 160px;
clear: left;}
.submit {
width: 310px;
text-align: right;}
</style>
</head>
<body>
<h1>Warsztaty poetyckie</h1>
<p>W tym roku mamy zamiar zorganizować i przeprowadzić serię warsztatów
oraz sympozjów poetyckich.</p>
<p>Dla naszych członków wstęp na te imprezy będzie bezpłatny:</p>
<ul>
<li>Perspektywa poetycka</li>
<li>Wojna w oczach Baczyńskiego</li>
<li>Poematy odnalezione i poeci-outsiderzy</li>
</ul>
<table>
<tr class="head">
<th></th>
<th>Kraków</th>
<th>Wrocław</th>
<th>Gdańsk</th>
</tr>
<tr>
<th>Perspektywa poetycka</th>
<td>Sobota, 4 lutego 2012<br />11:00 - 14:00</td>
<td>Sobota, 3 marca 2012<br />11:00 - 14:00</td>
<td>Sobota, 17 marca 2012<br />11:00 - 14:00</td>
</tr>
<tr class="even">
<th>Wojna w oczach Baczyńskiego</th>
```

# PRZYKŁAD

## LISTY, TABELY I FORMULARZE

```
<td>Sobota, 7 kwietnia 2012<br />11:00 - 13:00</td>
<td>Sobota, 5 maja 2012<br />11:00 - 13:00</td>
<td>Sobota, 19 maja 2012<br />11:00 - 13:00</td>
</tr>
<tr>
<th>Poematy odnalezione i poeci-outsiderzy</th>
<td>Sobota, 9 czerwca 2012<br />11:00 - 14:00</td>
<td>Sobota, 7 lipca 2012<br />11:00 - 14:00</td>
<td>Sobota, 21 lipca 2012<br />11:00 - 14:00</td>
</tr>
<tr class="even">
<th>Naturalna śmierć: Granice poznania</th>
<td>Sobota, 4 sierpnia 2012<br />11:00 - 16:00</td>
<td>Sobota, 8 września 2012<br />11:00 - 16:00</td>
<td>Sobota, 15 września 2012<br />11:00 - 16:00</td>
</tr>
</table>
<form action="http://www.jakaswitryna.com.pl/form.php" method="get">
<fieldset>
<legend>Rejestracja uczestnictwa</legend>
<p><label class="title" for="name">Imię i nazwisko:</label>
<input type="text" name="name" id="name"><br />
<label class="title" for="email">E-mail:</label>
<input type="text" name="email" id="email"></p>
<p><label for="location" class="title">Najbliższa placówka:</label>
<select name="location" id="location">
<option value="kr">Kraków</option>
<option value="wr">Wrocław</option>
<option value="gd">Gdańsk</option>
</select></p>
<span class="title">Czy jesteś członkiem stowarzyszenia?</span>
<label><input type="radio" name="member" value="yes" /> Tak</label>
<label><input type="radio" name="member" value="no" /> Nie</label></p>
</fieldset>
<div class="submit"><input type="submit" value="Rejestracja" /></div>
</form>
</body>
</html>
```



# PODSUMOWANIE

## LISTY, TABELE I FORMULARZE

- ▶ Oprócz właściwości CSS, które zostały opisane w innych rozdziałach i mogą być stosowane do określania wyglądu zawartości wszystkich elementów, dostępnych jest kilka właściwości przeznaczonych do określania wyglądu list, tabel i formularzy.
- ▶ Postać znaczników list można określać przy użyciu właściwości `list-style-type` oraz `list-style-image`.
- ▶ W różnych przeglądarkach komórki tabel mają różne obramowania oraz odstępy pomiędzy sobą; istnieją właściwości, które pozwalają kontrolować te aspekty wyglądu tabel i sprawić, by wyglądały spójnie.
- ▶ Korzystanie z formularzy będzie prostsze, jeśli ich pola zostaną idealnie wyrównane w pionie przy użyciu stylów.
- ▶ Formularze mogą zdecydowanie zyskać dzięki zastosowaniu stylów, które nadadzą im bardziej interaktywny wygląd.