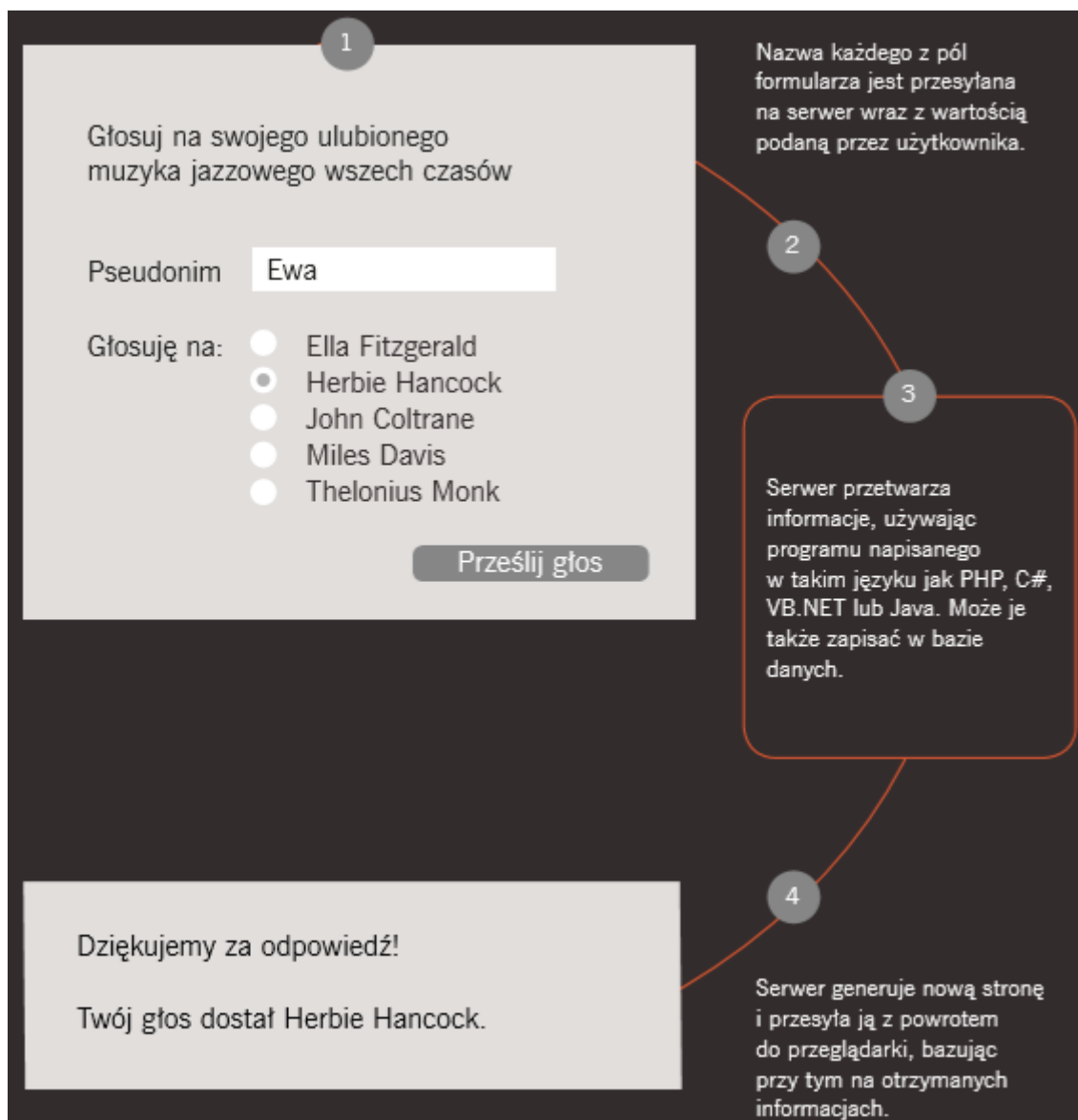




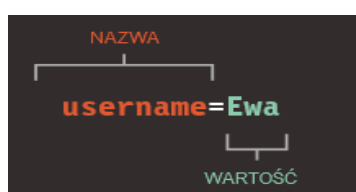
Jak działają formularze?

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 formularzem.

Aby odróżnić różne rodzaje wprowadzanych danych, informacje są przesyłane z przeglądarki a serwer w formie par nazwa=wartość.





Struktura formularzy

Wszystkie pola formularzy są umieszczone w złączniku `<form>`. W tym elemencie zawsze należy umieścić wartość atrybutu **action** oraz **method**.

action – każdy element `<form>` wymaga określenia wartości tego atrybutu. 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,
- formularzy służących do pobierania danych z serwera (a nie np. przesyłania informacji, które mają być zapisane lub usunięte z bazy danych).

W przypadku zastosowania metody POST wartości są przesyłane w tzw. nagłówkach HTTP. Metoda POST powinna być użyta, gdy formularz:

- pozwala a przesyłanie plików.
- jest bardzo duży.
- zawiera wrażliwe dane.
- powoduje dodanie informacji do bazy lub usunięcie informacji z bazy danych.

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

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9 <form action="localhost" method="POST">
10 <p>Tu będą umieszczone pola hormularzy</p>
11 </form>
12 </body>
13 </html>
```

Efekt:

Tu będą umieszczone pola hormularzy



Pola tekstowe

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 identyfikującą każde pole formularza.

maxlength atrybutu można używać w celu ograniczenia liczby znaków wpisywanych w polu. Jego wartość jest liczbą określającą, ile maksymalnie znaków można umieścić w danym polu.

size obecnie już rzadko nie stosowany. Dawniej wykorzystywany był do określania szerokości pola tekstowego do liczby znaków mających się tam zmieścić, obecnie szerokość pola ustawiana jest przy użyciu arkuszy CSS.

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9 <form action="localhost" method="POST">
10 <p>Nazwa użytkownika
11 <input type="text" name="username" maxlength="15">
12 </p>
13 </form>
14 </body>
15 </html>
```

Efekt:

Nazwa użytkownika



Pole hasła

<input>

type="password" dzięki temu atrybutowi można utworzyć pole działające podobnie do zwyczajnego, jednowierszowego pola tekstowego z tą różnicą, że wpisane do niego znaki będą maskowane.

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

size, maxlength znaczenie tych atrybutów jest takie samo jak w przypadku pola text.

Choć dane wpisywane w polu hasła są maskowane i nie można ich odczytać na ekranie, nie jest to równoznaczne z bezpiecznym przesyłaniem na serwer.

Aby zapewnić bezpieczeństwo należy w taki sposób skonfigurować serwer, aby komunikował się z przeglądarką używając protokołu SSL.

```
index.html x |
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9 <form action="localhost" method="POST">
10 <p>Nazwa użytkownika
11 <input type="text" name="username" maxlength="15">
12 </p>
13 <p>Hasło
14 <input type="password" name="password" maxlength="25">
15 </p>
16 </form>
17 </body>
18 </html>
```

Efekt:

Nazwa użytkownika

Hasło



Wielowierszowe pola tekstowe

<textarea>

Element służy do tworzenia wielowierszowych pól tekstowych. W odróżnieniu od elementu <input> musi mieć zarówno element otwierający, jak i zamykający.

W przypadku tworzenia nowych formularzy szerokość oraz wysokość elementu <textarea> należy określić z wykorzystaniem arkusza CSS. Jednak wielkość można również określić w samym złączniku.

cols określa szerokość pola (wyrażoną jako liczba znaków)

rows określa wysokość i wskazuje na liczbę wierszy tekstu, które mają być widoczne w polu.

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9 <form action="localhost" method="POST">
10 <p>Co myślisz o tym filmie?</p>
11 <textarea name="komentarz" cols="30" rows="4">
12 Tutaj wpisz swój komentarz...
13 </textarea>
14 </form>
15 </body>
16 </html>
```

Efekt:

Co myślisz o tym filmie?

Tutaj wpisz swój komentarz...



Pola jednokrotnego wyboru

<input>

- type="radio"** przyciski opcji pozwalają użytkownikom wybrać tylko jedną z grupy kilku opcji.
- name** atrybut określa nazwę pola. Jeśli odpowiedzi dotyczą tego samego pytania, to wszystkie muszą mieć taką samą wartość atrybutu name.
- value** atrybut określa wartość, jaka zostanie przesłana na serwer w razie zaznaczenia danej opcji. Wartości należące do jednej grupy powinny mieć unikatowe nazwy.
- checked** wskazuje przycisk opcji, który powinien zostać zaznaczony w momencie wyświetlenia strony. Można go umieścić tylko w jednym przycisku opcji w grupie.

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9 <form action="localhost" method="POST">
10 <p>Wybierz swój ulubiony rodzaj muzyki:
11 <br/>
12 <input type="radio" name="muzyka" value="rock" checked="checked">Rock
13 <input type="radio" name="muzyka" value="pop">Pop
14 <input type="radio" name="muzyka" value="jazz">Jazz
15 </p>
16 </form>
17 </body>
18 </html>
```

Efekt:

Wybierz swój ulubiony rodzaj muzyki:
 Rock Pop Jazz



Pola wyboru

<input>

type="checkbox" pola wyboru pozwalają na zaznaczenie dowolnej liczby odpowiedzi na zadane pytanie.

name określa nazwę pola. Grupa odpowiedzi do tego samego pytania muszą mieć taką samą wartość atrybutu name.

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

checked atrybutu można użyć, aby wskazać pole, które powinno zostać zaznaczone w momencie wyświetlenia strony.

```
index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5    <meta http-equiv="Content-Language" content="pl" />
6    <title>Tytuł strony</title>
7  </head>
8  <body>
9    <form action="localhost" method="POST">
10   <p>Wybierz swoją ulubioną usługę muzyczną:
11     <br />
12     <input type="checkbox" name="serwis" value="itunes" checked="checked">iTunes
13     <input type="checkbox" name="serwis" value="lastfm"> Last.fm
14     <input type="checkbox" name="serwis" value="spotify">Spotify
15   </p>
16 </form>
17 </body>
18 </html>
```

Efekt:

Wybierz swoją ulubioną usługę muzyczną:
 iTunes Last.fm Spotify



Listy rozwijane

<select>

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

Listy tego typu tworzone są przy użyciu elementu `<select>`, wewnątrz którego umieszczane są elementy `<option>`.

name określa nazwę pola.

<option>

Element służy do tworzenia opcji listy, z których użytkownik może wybierać. Słowa umieszczone pomiędzy otwartym znacznikiem `<option>`, a zamkniętym znacznikiem `</option>` będą wyświetlone na liście.

value określa wartość, która, wraz z nazwą pola zostanie przesłana na serwer po zaznaczeniu konkretnej opcji listy.

selected używany, by wskazać opcję, która ma być zaznaczona w momencie wyświetlania strony. W razie pominięcia tego atrybutu w polu zostanie wyświetlona pierwsza z podanych opcji. Jeśli użytkownik nie wybierze innej opcji, to w momencie wysłania formularza na serwer zostanie przesłana wartość pierwszej opcji.

Lista rozwijana pełni podobną funkcję do pól jednokrotnego wyboru. Pozwala na wybór jednej spośród wielu opcji.

```
index.html
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Jakiego urządzenia używasz do słuchania muzyki?</p>
12 <select name="urządzenia">
13 <option value="ipod">iPoda</option>
14 <option value="radio">radia</option>
15 <option value="komputer">Komputera</option>
16 </select>
17 </form>
18
19 </body>
20 </html>
```

Efekt:

Jakiego urządzenia używasz do słuchania muzyki?



Lista wielokrotnego wyboru

<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.

multiple umożliwia użytkownikom wybór kilku spośród opcji dostępnych na liście.

Aby zaznaczyć więcej niż jedną opcję należy wcisnąć i przytrzymać klawisz Ctrl.

```

1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <p>Czy grasz na następujących instrumentach? (
12 Możesz zaznaczyć więcej niż jedną opcję, naciskając klawisz Ctrl)</p>
13 <select name="instrumenty" size="3" multiple="multiple">
14 <option value="gitara" selected="selected">gitara</option>
15 <option value="perkusja">perkusja</option>
16 <option value="keyboard" selected="selected">keyboard</option>
17 <option value="gitara_basowa">gitara basowa</option>
18 </select>
19 </form>
20
21 </body>
22 </html>

```

Efekt:

Czy grasz na następujących instrumentach?
(Możesz zaznaczyć więcej niż jedną opcję, naciskając klawisz Ctrl)



Pole przesyłania plików

<input>

type="file"

zastosowanie tej wartości atrybuty type powoduje utworzenie pola umożliwiającego wyświetlenie okna dialogowego, pozwalającego na wybranie jednego z plików dostępnych na komputerze.

Uwaga!

Jeżeli formularz ma zapewniać możliwość przesyłania plików na serwer, to atrybut method elementu <form> musi mieć wartość POST. Plików nie można przysyłać przy użyciu metody GET.

Wyświetlane okno dialogowe dostosowane jest do używanego systemu operacyjnego i nie mamy możliwości jego zmiany.

```
index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <p>Prześlij plik muzyczny w formacie MP3:</p>
12 <input type="file" name="utwor"><br/>
13 <input type="submit" value="Prześlij">
14 </form>
15
16 </body>
17 </html>
```

Efekt:

Prześlij plik muzyczny w formacie MP3:



Przycisk przesyłający

<input>

type="submit" przycisk tego typu jest używany do przesyłania formularzy na serwer.

name określa nazwę pola.

value atrybut określa tekst, który będzie wyświetlany na przycisku.

Wygląd przycisku możemy określić w arkuszach CSS.

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Subskrybuj nasz biuletyn informacyjny:</p>
12 <input type="text" name="email">
13 <input type="submit" name="subskrypcja" value="Subskrybuj">
14 </form>
15
16 </body>
17 </html>
```

Efekt:

Subskrybuj nasz biuletyn informacyjny:



Przyciski z obrazem

<input>

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 .

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Subskrybuj nasz biuletyn informacyjny:</p>
12 <input type="text" name="email">
13 <input type="image" src="images/subskrybcja.jpg" width="100" height="20">
14 </form>
15
16 </body>
17 </html>
```

Efekt:

Subskrybuj nasz biuletyn informacyjny:



Element BUTTON oraz pola ukryte

<button>

Element zapewnia większą kontrolę nad wyglądem przycisków oraz pozwala na umieszczanie na nich innych elementów.

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

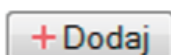
<input>

type="hidden" wartość atrybutu pozwalająca na tworzenie pól ukrytych.

Pola tego typu nie są widoczne na stronie. Pozwalają autorowi strony 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 przysyłania formularza.

```
index.html x
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <button>Dodaj</button>
12 <input type="hidden" name="bookmark" value="liryka">
13 </form>
14
15 </body>
16 </html>
```

Efekt:





Dodawanie etykiet do pól formularza

<label>

Każdemu z pól formularza powinien towarzyszyć element <label>. Element może być używany na dwa sposoby:

- wewnątrz niego można umieścić zarówno opis tekstowy, jak i samo pole formularza.
- można go oddzielić od pola formularza i jednocześnie dodać do niego atrybut `for` określający, z którym polem skojarzona jest dana etykieta.

for określa do którego pola formularza należy dana etykieta.

```
index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <label>Wiek:<input type="text" name="age"></label>
12 <br/>
13 Płeć:
14 <input type="radio" name="plec" value="k">
15 <label for="kobieta">kobieta</label>
16 <input type="radio" name="meczyczna" value="m">
17 <label for="meczyczna">mężczyzna</label>
18 </form>
19
20 </body>
21 </html>
```

Efekt:

Wiek:
Płeć: kobieta mężczyzna



Grupowanie elementów formularzy

<fieldset>

Powiązane ze sobą pola można zgrupować, umieszczając je wewnątrz elementu <fieldset>. Większość przeglądarek zawartość tych elementów wyświetla wewnątrz obramowania, co wizualnie grupuje powiązane ze sobą elementy. Wygląd obramowania można określić za pomocą stylów CSS.

<legend>

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

```

index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <fieldset>
12 <legend>Informacje kontaktowe:</legend>
13 <label>Adres e-mail:<br/>
14 <input type="text" name="email"></label><br/>
15 <label>Telefon komórkowy:<br/>
16 <input type="text" name="mobile"></label><br/>
17 <label>Telefon:<br/>
18 <input type="text" name="telefon"></label>
19 </fieldset>
20 </form>
21
22 </body>
23 </html>

```

Efekt:

Informacje kontaktowe:

Adres e-mail:

Telefon komórkowy:

Telefon:



Weryfikacja pól formularzy

Weryfikacja pól formularzy realizowana jest przy wykorzystaniu języka JavaScript. Jednak język HTML5 udostępnia mechanizm takiej weryfikacji.

Weryfikacja pól formularzy zapewnia, że po przesłaniu formularza na serwer informacje wpisane w polach zostaną prawidłowo zrozumiane.

Weryfikacja bazuje na zastosowaniu atrybutu **required**. Można go umieścić w dowolnym polu formularza, które użytkownik powinien wypełnić.

```
index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <label for="username">Nazwa użytkownika:</label>
12 <input type="text" name="username" required="required"><br/>
13 <label for="password">Hasło:</label>
14 <input type="password" name="password" required="required"><br/>
15 <input type="submit" value="Zaloguj się">
16 </form>
17
18 </body>
19 </html>
```

Efekt:

Nazwa użytkownika:

Hasło:

Zaloguj się

To jest pole wymagane



Pole daty

`<input>`

`type="date"` wartość atrybutu wyświetlająca pole służące do wyboru daty.

```
index.html x
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <meta http-equiv="Content-Language" content="pl" />
6  <title>Tytuł strony</title>
7  </head>
8  <body>
9
10 <form action="localhost" method="POST">
11 <label for="username">Data wylotu:</label>
12 <input type="date" name="depart">
13 <input type="submit" value="Prześlij">
14 </form>
15
16 </body>
17 </html>
```

Efekt:

Data wylotu:



Pola adresów e-mail i URL

<input>

type="email"

zastosowanie tej wartości atrybutu name sprawi, że przeglądarki będą sprawdzać, czy podane informacje zostały zapisane w prawidłowym formacie, charakterystycznym dla adresów poczty elektronicznej.

type="url"

zastosowanie tej wartości atrybutu name sprawi, że przeglądarki będą sprawdzać, czy podane informacje zostały zapisane w prawidłowym formacie, charakterystycznym dla adresów URL.

```

1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Podaj swój adres e-mail:</p>
12 <input type="email" name="email">
13 <input type="submit" value="Prześlij">
14 </form>
15
16 </body>
17 </html>

```

Podaj swój adres e-mail:

Należy wpisać poprawny adres e-mail

```

1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Podaj swój adres witryny:</p>
12 <input type="url" name="witryna">
13 <input type="submit" value="Prześlij">
14 </form>
15
16 </body>
17 </html>

```

Podaj swój adres witryny:

Należy wpisać poprawny adres URL



Pole wyszukiwania

<input>

type="search"

zastosowanie tej wartości atrybutu name udostępnia specjalne pole wyszukiwania.

placeholder

w dowolnym polu tekstowym można także użyć tego atrybutu. Jego wartość określa tekst, jaki będzie prezentowany w polu do momentu, gdy użytkownik je kliknie

```

1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Szukaj:</p>
12 <input type="search" name="szukaj">
13 <input type="submit" value="Szukaj">
14 </form>
15
16 </body>
17 </html>

```

Efekt:

Szukaj:


```

1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta http-equiv="Content-Language" content="pl" />
6 <title>Tytuł strony</title>
7 </head>
8 <body>
9
10 <form action="localhost" method="POST">
11 <p>Szukaj:</p>
12 <input type="search" name="szukaj" placeholder="Wpisz tekst...">
13 <input type="submit" value="Szukaj">
14 </form>
15
16 </body>
17 </html>

```

Efekt:

Szukaj: