



Nazwa implementacji: Formularze HTML

Autor:

Opis implementacji: Stworzymy dokument z formularzem umożliwiającym wprowadzanie informacji przez użytkownika z wykorzystaniem odpowiednich elementów w zależności od rodzaju informacji.

Formularze HTML służą zwykle do przesyłania informacji wprowadzanych przez użytkownika do aplikacji uruchomionej na serwerze (napisanej w takich językach jak np. PHP, Ruby, Python). Zamiast wysłać wprowadzone informacje na serwer, można przetwarzać je w przeglądarce używając JavaScriptu. Do takich aplikacji nie jest potrzebny serwer, a dzięki nowościom dostępnym w HTML5, możemy stworzyć wiele użytecznych aplikacji.

Podstawowy formularz

Najprostszy formularz HTML może składać się tylko z paru prostych tagów: `<form>` i `<input>` z odpowiednimi atrybutami:

```
<form>
  <input type="text" name="imie" />
  <input type="submit" value="Wyślij" />
</form>
```

Tag `<form>` zawiera tagi należące do formularza.

Tag `<input>` reprezentuje element formularza. Atrybut `type` może przyjmować różne wartości:

- `text` - tworzy pole tekstowe;
- `submit` - tworzy przycisk służący do wysyłania formularza do aplikacji na serwerze.

Oprócz atrybutu `type` wykorzystujemy również:

- `name` - określa nazwę zmiennej, pod którą wysłana zostanie wartość wprowadzona przez użytkownika;
- `value` - w przypadku `type="submit"` określa tekst znajdujący się na przycisku, a dla `type="text"` początkową wartość wprowadzoną w pole tekstowe.

Zadanie: Utwórz kopię pliku `index.html` z poprzedniej lekcji (w tym samym katalogu) i nazwij ją `formularz.html`. Zmodyfikuj linki nawigacyjne w obu dokumentach tak, aby możliwe było przejście z `index.html` na stronę `formularz.html` i z powrotem w przeglądarce. W pliku `formularz.html` zastąp zawartość elementu `.content` prostym formularzem HTML. Pamiętaj o dodaniu kilku parametrów CSS dla tagów `<form>` i `<input>` dla lepszego wyglądu strony.

Wprowadź imię i nazwisko

Wyślij

Etykiety



Przy większej ilości pól formularzu potrzebne są dodatkowe informacje dla użytkownika co ma być wprowadzane w poszczególne pola. Do tego wykorzystywany jest tag `<label>`:

```
<label for="imie">Imię i nazwisko</label>  
<input type="text" id="imie" name="imie" value="" />
```

Pomiędzy tagami `<label>` i `</label>` umieszczamy tekst, który ma znajdować się obok pola tekstowego, a poprzez atrybuty `for="imie"` i `id="imie"` łączymy je ze sobą.

Zadanie: Rozbuduj formularz o dodatkowe pola i ich opisy, korzystając z `<label>`.

Imię i nazwisko:

Email:

Uwaga: Dla łatwiejszego nadania takiego wyglądu formularza umieść pary tagów `<label>` i `<input>` dodatkowo w tagach `<div>` oraz skorzystaj z następującego kodu CSS:

```
div {  
  width: 450px;  
  margin: 0 auto 20px;  
  clear: both;  
}  
  
div input {  
  float: right;  
}
```

Dodatkowe atrybuty: `autofocus`, `placeholder`, `required`.

W HTML5 pojawiły się dodatkowe atrybuty pomagające w stworzeniu formularza, przyjaznego dla użytkownika:

- `autofocus` - umieszczenie tego atrybutu powoduje, że po wyświetleniu formularza przeglądarka automatycznie umieści kursor w danym elemencie, sugerując użytkownikowi miejsce rozpoczęcia wypełniania formularza.
- `placeholder` - służy do wprowadzania informacji pomocniczych, które znikają w momencie, gdy użytkownik zaczyna wypełniać dane pole.
- `required` - blokuje zatwierdzenie formularza, jeśli pole, w którym się znajduje, nie zostało wypełnione.

Zadanie: Uzupełnij formularz o nowe atrybuty, tworząc taki formularz tuż po wyświetleniu strony:



Imię i nazwisko:

Email:

a po kliknięciu „Wyślij”:

Imię i nazwisko:

Email:

Proszę wypełnić to pole.

Elementy formularza:

checkbox, radio, select, textarea

Formularze HTML nie składają się tylko z pojedynczych pól tekstowych i przycisków. Do dyspozycji mamy również listy jednorazowego i wielokrotnego wybory, rozwijalną listę, oraz pole tekstowe:

- `<input type="password">` - jest używany najczęściej do pól, które wymagają bezpiecznego wprowadzenia hasła;
- `<input type="hidden">` - nie jest widoczny w przeglądarce i służy do przesyłania dodatkowych informacji nie wprowadzanych bezpośrednio przez użytkownika;
- `<input type="checkbox">` - pozwala tworzyć pole, które użytkownik może zaznaczyć lub odznaczyć. Nazwę i wartość pola określamy poprzez **name** i **value**, a specjalny atrybut **checked**, pozwala oznaczyć te pola, które domyślnie mają być zaznaczone;
- `<input type="radio">` - działa na takiej same zasadzie jak checkbox ale pozwala użytkownikowi zaznaczyć tylko jedną odpowiedź;
- `<select>` - w przypadku gdy mamy dużo elementów na liście do wyboru i chcemy zaoszczędzić miejsca na stronie, możemy zamiast `<input type="radio">` skorzystać z tego tagu, a możliwe odpowiedzi umieścić wewnątrz jako tagi `<option>`:

```
<select name="miasto">
  <option value="">Wybierz miasto</option>
  <option value="waw" checked>Warszawa</option>
  <option value="krk">Kraków</option>
  <option value="poz">Poznań</option>
  ...
</select>
```

- `<textarea>` - używamy tak samo jak `<input type="text">` do wprowadzania tekstu, ale w momencie, kiedy oczekujemy, że użytkownik wprowadzi więcej niż tylko parę słów. Wielkość tego pola określamy obowiązkowymi parametrami `cols` i `rows`.





Zadanie: Uzupełnij formularz o nowe elementy:

Imię i nazwisko:

Email:

Hasło:

Mężczyzna Kobieta

Komentarz:

Wybierz miasto

Wyślij

Uwaga: Pamiętaj, że do tagów `<div>` możesz dodać atrybut `class`, który umożliwi nadania dla wybranej grupy tagów oddzielnych parametrów CSS. Na przykład:

```
<div class="radio">  
<input id="radio-m" type="radio" name="plec" value="m" />  
<label for="radio-m">Mężczyzna</label>  
<input id="radio-k" type="radio" name="plec" value="k" />  
<label for="radio-k">Kobieta</label>  
</div>
```

Możemy dodać parametry:

```
.radio {  
width: 260px;  
float: right;  
}  
.radio input {  
float: none;  
margin: 5px;  
}
```

Zaawansowane typy tagu

`<input>`

Uwaga: Poniższe wartości parametru `type` nie są aktualnie obsługiwane w najnowszej wersji przeglądarki Firefox. Można je jednak swobodnie stosować. Użytkownik z przeglądarką Firefox zobaczy zwykły element

4





`<input type="text">`.

Osoby zainteresowane mogą przetestować opisywane typy, korzystając z najnowszej wersji Chrome lub Opera. Więcej szczegółowych informacji o wsparciu w różnych wersjach przeglądarek jest dostępnych na stronach:

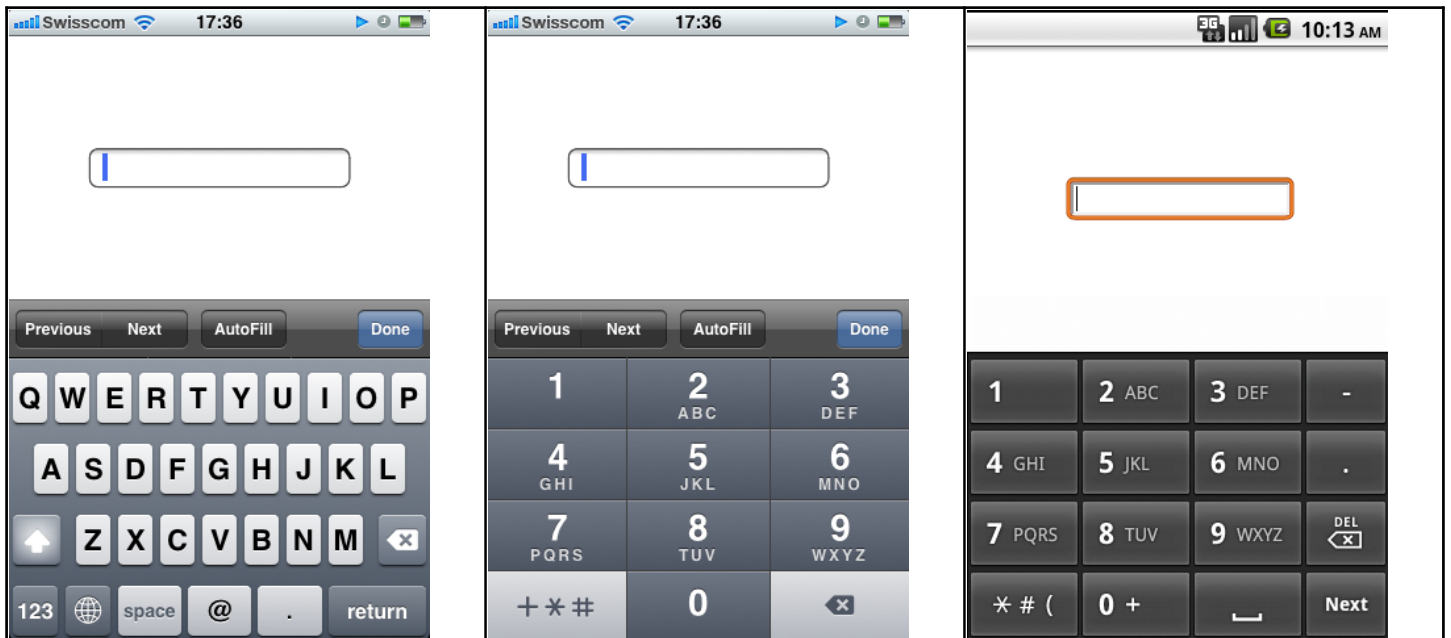
<http://caniuse.com/input-number>, <http://caniuse.com/input-range>, <http://caniuse.com/input-datetime>, <http://caniuse.com/input-color>.

Oprócz już poznanych wartości type dla tagu `<input>` HTML5 wprowadza kilka nowych:

- `number` – blokuje wprowadzanie innych danych niż wartości liczbowe; `range` – tak samo jak `number` tylko, że zamiast zwykłego pola wyświetlany jest suwak;

`date / time / color` –

służą do wprowadzania informacji o dacie, czasie lub kolorze; `email / tel / url` – nie zmieniają wyglądu i zachowania tagu `<input>` gdy korzystamy z przeglądarki desktopowej. Są wykorzystywane przez przeglądarki na urządzeniach mobilnych w celu dopasowania typu klawiatury, tak żeby ułatwić użytkownikowi wprowadzanie treści.



Źródło obrazków: <http://slides.html5rocks.com/#form-types-mobile>