

Technologie internetowe



CZĘŚĆ 2

HTML – WYMIANA DANYCH formularze

Wykład opracowany na podstawie materiałów z:

<http://pl.wikipedia.org/>

<http://www.kurshtml.edu.pl/>

<http://www.htmlkurs.yoyo.pl/index.html>

Czym są formularze ?



- Formularz HTML to front-end mechanizmu przekazywania danych do serwera WWW
 - Mogą też pełnić rolę prezentacyjną (Edycja danych)
- Funkcjonalność zależy od wymagań projektowych
 - Wysyłanie wiadomości do użytkowników
 - Rejestracja/logowanie do części zastrzeżonej serwisu
 - Określanie parametrów sterowania serwisem za pomocą zmiennych, preferencje użytkownika
 - Tworzenie kryteriów zapytań o dane zawarte w bazie danych
 - Itp..
- Wygląd oparty o kontrolki formularzy – dobór w zależności od potrzeb

Ramy formularza - atrybuty



- Ramy formularza znajdują się pomiędzy znacznikami
 - `<form> </form>`
- Znaczniki podstawowe posiadają atrybuty obowiązkowe i opcjonalne
 - **action** – atrybut obowiązkowy oznaczający gdzie mają zostać wysłane dane z formularza
 - ✦ Wartością atrybutu jest ścieżka http://nazwa_serwera.pl lub https://nazwa_serwera.pl zawierająca miejsce skryptu, który obsługuje dane formularza (php, asp, cgi...)
 - ✦ Można też dane wysłać bezpośrednio na adres poczty elektronicznej podając adres w formacie: mailto:jasio@nazwa_serwera.pl
 - ✦ Wysłanie e-maila wymaga skonfigurowania na serwerze usługi poczty wychodzącej
 - **method** – atrybut obowiązkowy, który definiuje metodę przekazywania danych do serwera
 - ✦ GET
 - ✦ POST

Ramy formularza - obsługa



- Skrypty typu server-side są w stanie wykonać działanie zapisu do zewnętrznego pliku (po stronie serwera) (*method GET/POST*)
- Skrypty typu client-side też mogą obsługiwać formularze lecz nie potrafią zapisywać danych do pliku
 - W tym przypadku podanie wartości dla parametru *method* nie jest konieczne

```
<form action="..." onsubmit="alert(this.tekst.value); return false"> <input type="text"
  name="tekst" />
  <input type="submit" value="OK" />
</form>
```

- Możliwość kontroli wprowadzonych danych przed ich wysłaniem

```
<form action="mailto:adres e-mail" method="post" onsubmit="if (this.tekst2.value == '')
  { alert('Zanim wyślesz formularz, musisz wypełnić pole tekstem!'); return false }">
  <input type="text" name="tekst2" />
  <input type="submit" value="OK" />
</form>
```

Ramy formularza - atrybuty



- **Znaczniki opcjonalne**
 - **name** – nazwa formularza wykorzystywana przez skrypty
 - **enctype** – parametr definiuje typ zawartości wysyłanej przez formularz
 - ✦ Domyślnie - *"application/x-www-form-urlencoded"*
 - ✦ *Tylko pola tekstowe - "text/plain".*
 - ✦ *Pliki - "multipart/form-data"*
 - ✦ *Inne zawartości – źródło: <http://filext.com> (Mime type)*
 - **target** – okno w którym ma się otworzyć plik (strona) do którego zostanie wysłany formularz

Ramy formularza - przykłady



```
<form action=„http://adres-serwera.pl/index.php?akcja=rejestruj”  
  method="post">
```

(Tutaj umieszcza się pola formularza)

```
</form>
```

```
<form action="mailto:jasio@adres-serwera.pl" method="post">
```

(Tutaj umieszcza się pola formularza)

```
</form>
```

```
<form action="mailto:adres e-mail?subject=temat" method="post">
```

(Tutaj umieszcza się pola formularza)

```
</form>
```

Ramy formularza



- Na stronie serwisu może istnieć kilka formularzy niezależnych
- Formularzy nie można tworzyć hierarchicznie (w ramach innego formularza)

```
<form name="form1dane"> ... .. </form>
... .. {zawartość strony} ... ..
<form name="form2ankieta"> ... .. </form>
```

- W formularzach wyróżnić można następujące elementy:
 - pole tekstowe
 - pole hasła
 - pole wyboru pliku
 - pole wyboru (wielokrotnego wyboru)
 - pole opcji (jednokrotnego wyboru)
 - przyciski akcji

```
<input type="typ_pola" />
```

Pole tekstowe



- Podstawowe pole danych w formularzach – służy do wprowadzania wartości ASCII

```
<input type="typ_pola" />
```

- **Atrybuty**
 - **value** – definiuje wartość domyślną pola (po wejściu na stronę pole będzie już wypełnione tą wartością). Wartość domyślną można zmienić.
 - **size** – określa szerokość pola tekstowego na stronie (szerokość podawana jest w ilości widocznych znaków. Użytkownik może wprowadzić dowolną liczbę znaków, parametr określa tylko ile znaków będzie widzianych).
 - **maxlength** – określa maksymalną ilość znaków jaką można wprowadzić do pola
 - **name** – określa nazwę własną pola, parametr służy jako nazwa zmiennej podczas wysyłania wartości pola do serwera.
 - **disabled** – blokada pola = „disabled”. Pola z tym parametrem nie są uwzględniane na tab-liście
 - **tabindex** - kolejność na liście pól dostępnych za pomocą klawisza tab. 0 – oznacza brak indeksacji, pozostałe liczby - kolejność

Pole hasła



- Pole służące do wprowadzania treści ASCII lecz ukrytej za znakami *. Najczęściej są to wartości haseł lub fraz, które muszą pozostać ukryte. Treść ukrywana jest jedynie na stronie. Przesyłana jest jako zwykły tekst bez szyfrowania.

```
<input type="password" name="nazwa" />
```

- Przesyłanie haseł zwykłymi formularzami nie jest bezpieczne. Lepiej wykorzystać bezpieczny protokół `https://`
- Atrybuty pól typu *password* są takie same jak pola typu *text*

Pole wyboru



- Pole wyboru jest typem używanym wtedy, gdy wartości wybierane przez użytkownika należą do ograniczonego zbioru

```
<input type="checkbox" name="nazwa" value="wartość" /> Tekst opisu pola
```

- Nazwa pola dotyczy np. pytania
- Wartość określa odpowiedź dla pytania (wartość dla pola)
 - ✦ **Wszystkie pola**, które dotyczą tego samego pytania muszą mieć **tą samą nazwę** ale **inne wartości** (dotyczy wysyłania danych e-mailem)
 - ✦ Jeśli pole obsługiwane jest przez JavaScript każde pole powinno posiadać unikatowy identyfikator `id=„..”`
 - ✦ Jeśli przesłane dane będą obsługiwane przez PHP – nazwy każdego z pól muszą być unikatowe (`nazwa[]` – wtedy odwołanie będzie można realizować na zasadzie `$nazwa($_POST['nazwa'])` lub `$_GET['nazwa']`)
 - ✦ Kolejne wartości będzie można odczytać jako:

```
nazwa[0], $nazwa[1], $nazwa[2] ($_POST['nazwa'][0], $_POST['nazwa'][1],  
$_POST['nazwa'][2])
```

lub

```
($_GET['nazwa'][0], $_GET['nazwa'][1], $_GET['nazwa'][2])
```

Pole wyboru



- Można zaznaczać jednocześnie wiele pól wyboru w formularzu
- Dodatkowe atrybuty dla pola wyboru:
 - **checked="checked"** - atrybut wartości domyślnej – zaznaczonej dla pola
 - **disabled="disabled"** – atrybut blokowania pola – nie można pola zaznaczyć ani odznaczyć
- Etykieta pola wyboru - opis nie jest związany logicznie z polem danych
- Połączenie etykiety z polem danych można wykonać za pomocą znacznika `<label> .. </label>`
- Pole może być zagnieżdżone w etykiecie lub nie

```
<label><input type="checkbox" name="nazwa" value="wartość" /> To jest etykieta pola obok</label>
```

```
<input type="checkbox" name="nazwa" value="wartość" id="pole1" /> <label for="pole1">To jest etykieta pola obok</label>
```

Pole wyboru



- Etykieta połączona logicznie z polem działa jak samo pole danych. Kliknięcie na etykietę powoduje zaznaczenie lub odznaczenie pola.
- Druga z propozycji połączenia logicznego wykorzystuje się w przypadku lokowania formularza w tabeli
- Etykiet nie można zagnieżdżać (etykieta w etykiecie)
- Klawisz skrót służy do przeniesienia sterowania na dane pole.
- Tworzy się go za pomocą atrybutu **accesskey="... "**
- Egzekwuje się za pomocą Alt+Przypisany_klawisz (np. a) w Opera – Shift+ESC+Przypisany_Klawisz

Pole wyboru - blokowanie innego pola



- Czasem trzeba by pole było aktywne dopiero kiedy zaznaczenie innego pola na to pozwoli (np. występuje lista opcji i jako dodatkową wartość dopuszcza się treść wpisaną przez użytkownika)

```
<input type="checkbox" name="nazwa" value="wartość"
onclick="this.form.elements['nazwa2'].disabled = this.checked" />
<input type="text" name="nazwa2" />
```

- Lub odwrotnie pole tekstowe jest zablokowane po zaznaczeniu opcji

```
<input type="checkbox" name="nazwa" value="wartość"
onclick="this.form.elements['nazwa2'].disabled = this.form.elements['nazwa3'].disabled
= this.checked" />
<input type="text" name="nazwa2" />
<input type="text" name="nazwa3" />
```

Pole wyboru - ukrywanie innego pola



- Odkrywanie lub ukrywanie innego pola można uzależnić od pole wyboru

- Odkrywanie pola

```
<input type="checkbox" name="nazwa" value="wartość"
  onclick="document.getElementById('identyfikator').style.display =
  this.checked ? 'block' : 'none'; this.form.elements['nazwa2'].disabled
  = this.form.elements['nazwa3'].disabled = !this.checked" />
<div id="identyfikator" style="display: none">
<input type="text" name="nazwa2" disabled="disabled" />
<input type="text" name="nazwa3" disabled="disabled" />
</div>
```

- Ukrywanie pola

```
<input type="checkbox" name="nazwa" value="wartość"
  onclick="document.getElementById('identyfikator').style.display =
  this.checked ? 'none' : 'block'; this.form.elements['nazwa2'].disabled
  = this.form.elements['nazwa3'].disabled = this.checked" />
<div id="identyfikator">
<input type="text" name="nazwa2" />
<input type="text" name="nazwa3" />
</div>
```

Pole opcji

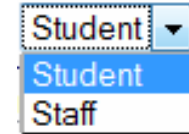


- Pole opcji jest typem, który wybiera jedną wartość z wielu.

```
<input type="radio" name="nazwa" value="wartość" />opis pola
```

- Pola tego samego pytania (opcji) muszą mieć tą samą nazwę zaś wartości dla pól koniecznie muszą być inne
- W przypadku obsługi przez JavaScript każde z pól powinno mieć inne, unikatowe w skali strony *id=".."*
- Pole opcji jest okrągły polem – stosowane w większej liczbie w formularzu można wybrać tylko jedno pole
- Atrybuty dodatkowe:
 - `checked= "checked"` – wartość domyślna (zaznaczająca pole)
 - `disabled= "disabled"` – wartość blokująca pole
- Dla pola opcji można również przypisać etykietę oraz klawisz skrótu jak dla pola wyboru

Lista rozwijalna (combi)



A dropdown menu with a blue border. The selected item is 'Student' in white text on a blue background. Below it, 'Student' and 'Staff' are listed in blue text on a white background.

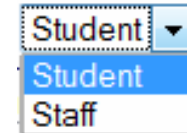


- Pole dysponujące wartościami opisowymi w postaci rozwijalnej listy

```
<select name="nazwa">  
<option>Tu wpisz pierwszą możliwość</option>  
<option>Tu wpisz drugą możliwość</option>  
(...) </select>
```

- Nazwa – określa nazwę pola danych
- Każda wartość `<option> dane </option>` to kolejna dana w polu. W *danych* nie można użyć żadnych znaczników
- Aby zostały wysłane inne wartości niż znaczniki stosuje się atrybut `value= "wartość"`. Atrybuty te są obowiązkowe jeśli zamierzamy wykorzystać wartość w skryptach
- W polu można wybrać jedną (tryb podstawowy) lub kilka (Ctrl) wartości dodając do `<select>` atrybut **multiple="multiple"**
- Aby odebrać wszystkie zaznaczone opcje w skrypcie PHP jako nazwę pola można wpisać `nazwa="nazwa[]"`

Lista rozwijalna (combi)

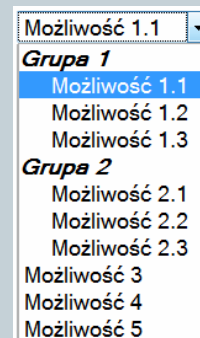


Student ▼
Student
Staff



- Wtedy dostęp do wartości będzie realizowany przez zmienną `$nazwa ($_POST['nazwa'] lub $_GET['nazwa'])`
- Wartości zaznaczonych opcji będą odczytywane jako `$nazwa[0]`, `$nazwa[1]`, `$nazwa[2]` (`$_POST['nazwa'][0]`, `$_POST['nazwa'][1]`, `$_POST['nazwa'][2]` albo `$_GET['nazwa'][0]`, `$_GET['nazwa'][1]`, `$_GET['nazwa'][2]`)
- Opcje domyślne posiadają atrybut `<option selected="selected" >`
- Aby można było zaznaczyć kilka opcji domyślnych należy użyć atrybutu **multiply**
- Określenie wysokości pola ustala się atrybutem **size="w"**, gdzie w to liczba widocznych linijek w polu
- Blokada pola **disabled="disabled"**
- Grupowanie opcji

```
<optgroup label="nazwa grupy">  
<option>Tu wpisz pierwszą możliwość</option>  
<option>Tu wpisz drugą możliwość</option>  
</optgroup>
```



Możliwość 1.1 ▼
Grupa 1
Możliwość 1.1
Możliwość 1.2
Możliwość 1.3
Grupa 2
Możliwość 2.1
Możliwość 2.2
Możliwość 2.3
Możliwość 3
Możliwość 4
Możliwość 5

Obszar tekstowy

Tego tekstu nie
można usunąć ani
zmienić



- Jest polem tekstowym, które pozwala na wprowadzenie danych w bloku (jak w edytorach)

```
<textarea name="nazwa" cols="x" rows="y">
```

Treść zawarta w polu jako wartość domyślna

```
</textarea>
```

- Nazwa – oznacza nazwę pola wysyłaną z formularze
- X, Y, - są liczbami wyrażającymi liczbę kolumn i wierszy jednocześnie widocznych w polu *textarea*
- Pomędzy znacznikami *<textarea>* i *</textarea>* nie można wpisywać żadnych znaczników
- Klawisz *Enter* przechodzi od następnej linijki w polu. Uwzględniane są tabulatory i kilkukrotne spacje.

Obszar tekstowy

Tego tekstu nie
można usunąć ani
zmienić



- **Dodatkowe opcje:**
 - `readonly` = „readonly” - tekst nie będzie mógł być modyfikowany
 - `disabled` = „disabled” – blokuje pole jak w poprzednich przykładach
 - `wrap` = „rodzaj” – określa typ zawijania tekstu (jest wartością zdeprecjonowaną)
 - ✦ „off” – wyłączenie zawijania
 - ✦ „physical” zawijanie tekstu na stronie i przesyłanie w formularzu tekstu zawiniętego
 - ✦ „virtual” – zawijanie tekstu na stronie, przesyłanie tekstu w rozwiniętych liniijkach.

Selektory plików

Przełdaj...



- Jest to pole za pomocą którego można wybrać plik i przesłać go formularzem

```
<form action="..." enctype="multipart/form-data">  
<input type="file" name="nazwa" />  
</form>
```

- Nazwa określa nazwę pola przesłaną z formularzem
- Jeśli nie zostanie podana ścieżka: multipart/form-data to wartością przesłana przez formularz nie będzie plik a tylko tekst (lokalnej) ścieżki dostępu. (obsługa PHP)
- Ważne by określić rodzaj akceptowalnych plików przesyłanych na serwer – **BARDZO NIEBEZPIECZNE !!!**
 - `<form action="..." enctype="multipart/form-data" accept="lista typów">`
 - `<input type="file" name="nazwa" accept="lista typów" />`
- Typy plików można określać globalnie dla całego formularza lub lokalnie dla każdego z selektorów.
 - `<input type="file" name="..." accept="image/jpeg,image/gif" />`

Selektory plików

Przełącznik



- Typ pliku w jednym systemie może być używany do czego innego niż w drugim stąd wykorzystuje się z typów MIME:
 - *.gif image/gif Graphic Interchange Format
 - *.htm text/html HyperText Markup Language
 - *.jpg image/jpeg JPEG
 - [Pełny zestaw typów MIME](#)
- Typ pliku oraz jego dopuszczalną wielkość można również określić w pliku php.ini

Dane ukryte



- Element służy do utworzenia pola, które przechowuje wartość i przesyła w formularzu ale nie jest widoczne

```
<input type="hidden" name="nazwa" value="wartość" />
```

- Nazwa – określa nazwę pola
- Wartość – przechowywana wartość w polu
- Pole jest dość powszechnie wykorzystywane do przesyłania między klientem a serwerem danych synchronizujących

Wysyłanie formularza

Prześlij zapytanie



- Jest to przycisk, służący do rozpoczęcia akcji przesyłania danych formularza

```
<input type="submit" value="wartość" />
```

- Wartość – określa napis, który pojawi się na przycisku

- Dodatkowe opcje:

- disabled=„disabled” – blokuje pole

- Przycisk obrazkowy

```
<input type="image" src="ścieżka dostępu do obrazka" alt="informacja alternatywna" />
```

- src=„ścieżka pliku obrazka” – podaje ścieżkę do pliku, który ma być obrazkiem zastępującym klasyczny przycisk.

- alt=„informacja tekstowa” – tekst pojawiający się „w dymku”

- Dodatkowe własności – można przesyłać wartość naciśniętego przycisku

```
<form action="?akcja=kasowanie" method="post">
<input type="hidden" name="id" value="1" />
<p> Czy na pewno chcesz to zrobić? </p>
<input type="submit" name="potwierdzenie" value="Tak" />
<input type="submit" name="potwierdzenie" value="Nie" />
</form>
```

Czyszczenie danych

Resetuj



- Element formularza tworzy przycisk za pomocą którego wartości wszystkich pól są usuwane
 - `<input type="reset" value="wartość" />`
- Wartość – oznacza treść prezentowaną na przycisku
- Dodatkowe atrybuty:
 - `disabled="disabled"` – blokowanie pola
- Potwierdzenie wyczyszczenia pól formularza
 - W znaczniku `<form>` należy dodać obsługę JavaScript:
`onreset="if (!confirm('Czy na pewno chcesz wyczyścić cały formularz?')) return`

Przyciski formularza



- Element służy do wprowadzenia do formularza przycisku dla którego określić można nietypową (submit, reset) akcję: *Type=„button”*
- Przyciskiem może być grafika, tabelka i inne elementy stojące pomiędzy znacznikami *<button>* a *</button>*
- Akcję dla przycisku określić należy za pomocą kodu *JavaScript* w akcji *onclick*

Przykład z tabelką:

komórka1 komórka2

komórka3 komórka4

Grupowanie pól



- Pozwala na tworzenie graficznych granic grupowania pól w formularzu

```
<fieldset>
```

```
<legend align="rodzaj">Tytuł</legend>
```

```
  (Pola formularza)
```

```
</fieldset>
```

Wybierz opcję:

Opcja 1

Opcja 2

Opcja 3

Wpisz dane

Polskie znaki



- Często problemem w wysłaniu formularzy są polskie znaki diakrytyczne
- Jest wiele sposobów usuwania polskich znaków szczególnie w nazwach plików
- Jednym jest skrypt JavaScript w połączeniu z danymi formularza
- [Do wglądu na stronie samouczka Helion](#)
- Strona kodowa formularza
- Zmiana strony kodowej formularza następuje po wprowadzeniu do `<form>` argumentu

`accept-charset=„strona kodowa”`

- Ten parametr ustala docelowy format w którym dane zostaną przesłane do przetworzenia
- [Przykład formularza](#)

Netografia



- Generator pól formularza:

<http://www.kurshtml.edu.pl/generatory/form.html>

Pytania sprawdzające



- Jaką rolę w serwisach pełnią formularze?
- Od czego zależy funkcjonalność formularzy?
- Rama formularza i najważniejsze atrybuty?
- Co to jest metoda GET i na czym polega?
- Co to jest metoda POST i na czym polega?
- Na czym polega obsługa danych z pól formularza?
- Jakie przykładowe technologie są przydatne w obsłudze danych z formularzy?
- Wymienić rodzaje pól danych w formularzu?
- W jaki sposób przesyła się dane z formularza na serwer?
- Jaki atrybut odpowiada za nazwę zmiennej w skrypcie obsługi?
- Sposoby uruchomienia formularza?
- Dodatkowe atrybuty w polach formularza?
- Na czym polega grupowanie pól w formularzu?

Koniec



Temat następnego wykładu to:

CSS