

Zadanie z PHP - 18 kwi 2012

18 kwietnia 2012

Spis treści

1 Część statyczna – HTML	1
1.1 Formularz	2
2 Kod PHP	2
2.1 Funkcje	3
2.2 Czytanie pliku przez PHP	3
2.3 Wyjaśnienie kodu	4
3 Różności o różnościach	4
3.1 Metody wysyłania formularzy	4
3.2 Tryby otwierania pliku	5

1 Część statyczna – HTML

W pliku `zapytaj.php` znajduje się szablon HTML (to zwyczajne, czyli `doctype`, `head`, `body` itd.) oraz kod poniżej:

```
1 <form action="./odpowiedz.php" method="post">
2   <label for="ssak">Ssak</label>
3   <input type="text" size="30" maxlength="100" id="ssak" name="ssak" />
4
5   <label for="rozmiar">Rozmiar</label>
6   <input type="text" size="10" maxlength="10" id="rozmiar" name="rozmiar" />
7
8   <label for="liczba">Liczba</label>
9   <input type="text" size="12" maxlength="12" id="liczba" name="liczba" />
10
11   <input type="submit" value="Zapytaj" />
12 </form>
```

Pochylmy się nad poszczególnymi linijkami, aby dowiedzieć się, co każda z nich robi.

1.1 Formularz

Znacznik form Tenże znacznik oznacza, że w nim znajduje się formularz. Atrybut `action` definiuje do jakiego pliku ma zostać wysłana zawartość formularza (tam musi się znajdować skrypt z odpowiedzią), a `method` – jaką **metoda**.

Znacznik label służy do oznaczania etykiet dla pól formularza. Po prostu – taki znacznik istnieje, więc się go używa. Jako atrybut możemy dać `for`, w które wpisujemy identyfikator pola, dla którego etykieta jest przeznaczona.

Znacznik input to zwyczajnie pole do wprowadzania. Różne są jego rodzaje (określane przez `type`): `text`, `email`, `submit`, `reset`. . . My wykorzystujemy głównie `text` – służy do wpisywania tekstu oraz `submit` – przycisk do wysłania formularza.

Text Dla tego pola możemy nadać przeróżne atrybuty, oto najważniejsze:

- `size` – długość pola mierzona w znakach,
- `maxlength` – maksymalna długość ciągu jaki można wpisać w pole,
- `id` – identyfikator pola,
- `name` – nazwa pola, wykorzystywana później w pliku odpowiedzi.

Submit Najważniejszym atrybutem jest `value`, które zwyczajnie mówi jaki tekst ma się wyświetlić na przycisku.

2 Kod PHP

```
1 <?php
2 $result = "<p>Nie_ma.</p>";
3 $file = fopen("nazwy.txt", "r");
4 while(!feof($file))
5 {
6     if(trim(fgets($file)) == trim($_POST['ssak']) && isset($_POST['ssak']) && $_POST['ssak'] != '')
7     {
8         $result = "<p>OK.</p>";
9     }
10 }
11 fclose($file);
12 echo $result;
13 if(is_integer($_POST['rozmiar']) && $_POST['rozmiar'] >= 10 && $_POST['rozmiar'] <= 100)
14 {
15     echo "<p>OK.</p>";
```

```

16 }
17 else
18 {
19     echo "<p>Rozmiar_spoza_zakresu.</p>";
20 }
21 $file = fopen("liczby.txt", "a");
22 fwrite($file, "\n" . $_POST['liczba']);
23 fclose($file);
24 echo "<p>".$_POST['rozmiar']*$_POST['liczba']."</p>";
25 ?>

```

Przypomnijmy sobie najpierw o co chodzi w zadaniu. Mamy sprawdzić, czy w pliku `nazwy.txt` istnieje nazwa gatunku wysłana przez użytkownika, sprawdzić parę rzeczy jeśli chodzi o rozmiar podany w formularzu, a na koniec dopisać liczbę podaną przez użytkownika do pliku `liczby.txt`.

2.1 Funkcje

W kodzie tym znajduje się dużo interesujących funkcji, więc najpierw opiszę co każda z nich robi, a później będziemy oglądać na przykładzie tegoż pliku:

`fopen(nazwa-pliku, tryb)` Otwiera plik *nazwa-pliku*. Musimy zapisać to co funkcja zwraca do zmiennej, aby móc później korzystać z tego otwartego pliku. Aby poczytać o różnych trybach otwierania pliku, [kliknij tu](#).

`feof(otwarty-plik)` Sprawdza, czy wskaźnik czytający plik doszedł już do końca pliku. Zwraca `true` lub `false`.

`fgets(otwarty-plik)` Czyta plik linia po linii, przesuwając wskaźnik. Zwraca kolejną linię pliku.

`fclose(otwarty-plik)` Zamyka plik, zapisuje zmiany.

`fwrite(otwarty-plik, tekst-do-wpisania)` Dopisuje do *otwartego-pliku* *tekst-do-wpisania*.

`trim(ciąg-znakow)` Usuwa „białe znaki” z początku i końca ciągu znaków. Białe znaki to spacje, znaki nowej linii i takie inne.

`isset(nazwa-zmiennej)` Sprawdza, czy zmienna o danej nazwie została już gdzieś zdefiniowana.

`echo(tekst)` Wypisuje tekst.

`is_integer(zmienna)` Sprawdza, czy zmienna jest liczbą całkowitą. Zwraca `true` lub `false`.

2.2 Czytanie pliku przez PHP

Zacznijmy od tego tematu. Gdy PHP musi przeczytać jakiś plik, ustawia sobie najpierw wskaźnik, a następnie przesuw go w te, czy wewte. Wskaźnik – uznajmy to za metaforyczną kreseczkę, jaką się widuje w edytorach tekstu. Gdy PHP otwiera plik w trybie „r” ustawia sobie wskaźnik na początek pliku, a następnie gdy wywołujemy funkcję `fgets` bierze wskaźnik i przesuwa go coraz to dalej, aż natrafi na znak nowej linii. Wyobraźmy to sobie jako przesuwanie strzałką w prawo kursora w tekście. Gdy znajdzie koniec linii, całą linię kopiuje do schowka, a później wkleja nam i wypłuwa.

2.3 Wyjaśnienie kodu

Stąd taka konstrukcja kodu. Najpierw otwieramy odpowiedni plik w odpowiednim trybie, a później dopóki wskaźnik nie doszedł do końca pliku (`fEOF`) pobieramy kolejne linie (`fgets`) i sprawdzamy, czy nie są równe gatunkowi ryby. Używamy tam funkcji `trim`, ale czemu. Otóż mam wrażenie, że `fgets` podaje linię z pliku razem ze znakiem nowej linii na końcu (której nie ma w danej z formularza), więc tą funkcją zwyczajnie usuwamy zbędną linię. Dodatkowo sprawdzamy, czy w ogóle formularz został wysłany i czy podany gatunek nie jest pusty. Takie zabezpieczające linie. No i jeśli linia z pliku równa jest gatunkowi, to odkryliśmy, że podany gatunek jest w pliku.

Kolejna linia sprawdza warunki na podanym rozmiarze.

1. Sprawdza, czy jest liczbą całkowitą.
2. Sprawdza, czy jest większe lub równe 10.
3. Sprawdza, czy jest mniejsze lub równe 100.

Jeśli liczba spełnia te warunki – wypisujemy OK. W przeciwnym wypadku co innego.

No i na koniec przechodzimy do dodania liczby do pliku tekstowego. Podobnie jak przy wczytywaniu:

1. Otwieramy plik w odpowiednim trybie.
2. Dopisujemy do niego ciąg `"\n".$_POST['liczba']`, czyli znak nowej linii połączony z liczbą.
3. Zamykamy edycję pliku.

No rzecz ostatnia – wypisujemy pomnożone rozmiar i liczbę.

3 Różności o różnościach

3.1 Metody wysyłania formularzy

Istnieją dwie metody wysyłania formularzy – GET i POST. Różnica jest prosta.

Dane formularza wysyłane metodą GET są dodawane do linku. Czyli, jeśli formularz wysyłamy metodą GET, a mamy w nim pole tekstowe „długość”, w które wpisaliśmy 144, to po kliknięciu przycisku wysłania formularza (zakładając, że `action="./response.php`) przejdziemy na stronę `http://costam/folder/response.php?długość=144`.

Dla formularza:

```
<form action="./odpowiedz.php" method="get">
    <input type="text" name="interesujace-pole" /> <!-- Wyp. zawartosc1 -->
    <input type="text" name="nieinteresujace-pole" /> <!-- Wyp. zawartosc2 -->
    <input type="submit" value="Wyślij, proszę. :)" />
</form>
```

Po naciśnięciu przycisku przejdziemy na stronę

`plepleple/odpowiedz.php?interesujace-pole=zawartosc1&nieinteresujace-pole=zawartosc2`.

W pliku odpowiedzi możemy dostać tą wartość za pomocą zmiennej `$_GET['nazwa-pola']`. Jak choćby w powyższym przypadku mamy dostęp do `$_GET['interesujace-pole']` oraz do `$_GET['nieinteresujace-pole']`.

Jeśli chodzi natomiast o metodę POST – dane wysyłane są „tajemnymi drogami”, nie widzimy ich w adresie strony. A z poziomu skryptu odpowiedzi korzystamy z tablicy `$_POST`. Jak choćby:

```
<form action="./odpowiedz-inna-a-co.php" method="post">
    <input type="text" name="laaaaaal" /> <!-- Wyp. zawartoscnumerjeden -->
    <input type="text" name="dziaala" /> <!-- Wyp. zawartoscnumerdwa -->
    <input type="submit" value="Wyślij, proszę. :)" />
</form>
```

W pliku `odpowiedz-inna-a-co.php` mamy dostęp do dwóch zmiennych:

- `$_POST['laaaaaal'] = 'zawartoscnumerjeden'`
- `$_POST['dziaala'] = 'zawartoscnumerdwa'`

3.2 Tryby otwierania pliku

Można różnie otwierać plik w PHP. Dzięki temu wiemy, że np. otwierając ten plik nie nadpiszemy sobie nic, że przypadkiem ktoś niepowołany nie przeczyta go itd. A oto jak wyglądają i co robią poszczególne tryby:

r	Pozwala na czytanie pliku. Wskaźnik na początku pliku.
r+	Pozwala na czytanie i zapisywanie w pliku. Wskaźnik na początku.
w	Pozwala na zapisywanie do pliku. Jeśli plik istnieje – otwiera go i czyści. Jeśli nie istnieje – tworzy nowy.
w+	Pozwala na czytanie i zapisywanie pliku. Jeśli plik istnieje – otwiera go i czyści. Jeśli nie istnieje – tworzy nowy.
a	Pozwala na dodawanie tekstu do istniejącego pliku. Jeśli plik istnieje – otwiera go i wskaźnik daje na koniec. Jeśli nie istnieje – tworzy nowy.
a+	Pozwala na dodawanie tekstu do istniejącego pliku oraz czytanie jego zawartości. Jeśli plik istnieje – otwiera go i wskaźnik daje na koniec. Jeśli nie istnieje – tworzy nowy.
x	Pozwala na zapisywanie do pliku. Tworzy nowy.
x+	Pozwala na zapisywanie do pliku i czytanie jego zawartości. Tworzy nowy.