



# TIN

## Skrypt 7.

PJWSTK, 2011

Marcin Dąbrowski

[mdabrowski@pjwstk.edu.pl](mailto:mdabrowski@pjwstk.edu.pl)



# Wprowadzenie

PHP (PHP Hypertext Preprocessor) jest językiem skryptowym (zagnieżdżanym w HTML), wykonywanym po stronie serwera. Rozwiązanie takie, określane jako styl SSI (Server Side Includes – od historycznie wcześniejszej technologii zakładającej wstawianie do strony serwerowych elementów skryptowych), jest obecnie najpopularniejsze z uwagi na jego produktywność.

**Cel przetwarzania strony PHP = wyprodukowanie na żądanie klienta dokumentu HTML.**

Ułatwienie polega na tym, że zamiast programować generowanie całej treści strony, można jej statyczne elementy napisać jak statyczny HTML, zaś dynamiczne – wkomponować w postaci skryptu w tę strukturę..

## Ćwiczenia

### ĆWICZENIE A – PODSTAWY PHP

PRE: Wszystkie ćwiczenia z obecnego skryptu można przećwiczyć przy pomocy szkolnego serwera WWW. W tym celu należy utworzyć plik test.php i umieścić w folderze public\_html na dysku Z:. Sprawdzenie działania strony możliwe jest przez przeglądarkę z adresu:

[www.users.pjwstk.edu.pl/~s9999\[numer\\_albumu\]/test.php](http://www.users.pjwstk.edu.pl/~s9999[numer_albumu]/test.php)

1. Sprawdzamy działanie procesora PHP: tworzymy prostą stronę HTML (nagłówek i ciało), jednak zapisujemy ją z rozszerzeniem .php. W obszarze BODY umieszczamy skrypt PHP, „opakowany” w następujący znacznik: `<?php ... ?>`. Wewnątrz umieszczamy polecenie `echo 'Dowolny tekst';`
2. Zadeklaruj zmienną i sprawdź jej użycie:
  - wyświetlenie zmiennej w tekście polecenia echo;
  - instrukcja warunkowa oparta o wartość tej zmiennej;
3. Zapoznaj się z różnicą pomiędzy stosowaniem w poleceniu echo „cudzysłów” i „apostrofów”. Wprowadź jeszcze jedną instrukcję `if (...){ }else { }`, tym razem przeplatając ją statycznym HTML (tzn. zamykamy znacznik skryptu przed zakończeniem instrukcji, a dalej otwieramy nowy dla jej dalszej części).
4. Spróbuj też wstawić znacznik HTML (np. `<I>...</I>`) do wnętrza łańcucha wyświetlanego komendą echo.
5. Wyświetl aktualny czas, korzystając z funkcji `date()`. Wymaga ona parametru określającego format daty – np. H:i. Pełna lista poleceń formatujących:  
<http://pl.php.net/manual/pl/function.date.php>
6. Korzystając z ww. poleceń wyświetl datę w postaci 2011.12.31. Poniżej, wyświetl czas w postaci „9:33 [am]”.



## ĆWICZENIE B – FORMULARZE I PRZESYŁANIE DANYCH

1. Sporządź formularz HTML zawierający 2 lub 3 pola tekstowe (np. z danymi osobowymi) oraz przycisk Submit. Ustal zachowanie „Submit” (atrybut „action”) jako otwarcie nowej strony: wynikZgloszenia.php z przekazaniem (metodą GET lub POST) zawartości formularza.
2. Budowę strony wynikZgloszenia.php rozpocznij od pustego szkieletu dokumentu HTML zawierającego nagłówek: „Potwierdzenie zgłoszenia”. Początkowo w powyższym formularzu wybierz tryb GET, aby już przy takim „pustym” dokumencie być w stanie sprawdzić, czy parametry są właściwie przekazywane (wyświetlą się w adresie URL w przeglądarce).
3. Rozbuduj przykład do typowej rejestracji podstawowych danych osobowych. Postaraj się przećwiczyć wszystkie podstawowe kontrolki formularza.
4. Zapoznaj się ze sposobem sięgania do przekazanych parametrów z poziomu skryptu i wyświetl wyniki w postaci podsumowania rejestracji użytkownika.

## ĆWICZENIE C – INCLUDE I PLIKI

1. Ważną możliwością dla skutecznego zarządzania treścią i kodem jest funkcja importowania (wstawiania) zawartości innego pliku. Służy temu funkcja require(nazwaPliku). Użyj ww. funkcji dla ujednoczenia np. górnego paska swoich stron. Projekt zawartości tych wstawianych dostosuj do dokładnego miejsca w strukturze strony, w którym chcesz je umieszczać.
2. Sprawdź wywołanie własnej funkcji napisanej w php a zadeklarowanej w osobnym pliku .inc
3. Przygotuj plik linki.txt, w którym umieścisz (w kolejnych wierszach) kilka wpisów w postaci: adres URL strony, średnik, krótka nazwa strony mogąca zawierać spacje. Niech przy każdym wczytaniu strony wykonywane będzie losowanie linku do wyświetlenia, czyli:
  - załadowanie tego pliku do tablicy
  - wylosowanie indeksu w tej tablicy (funkcja rand(dolZakresu, goraZakresu) ).
  - wygenerowanie znacznika A z odpowiednią wartością atrybutu HREF oraz odpowiednim tekstem wewnątrz znacznika.
4. Uzupełnij mechanizm o możliwość przekazania parametru. Niech żądanie test.php?id=5 generuje stronę dla piątego linku w pliku.

## ĆWICZENIE D – GALERIA

1. Przygotuj stronę internetową która będzie potrafiła wyświetlić wszystkie zgromadzone na serwerze obrazy. Stwórz stronę galery.php, która podczas generowania wynajdzie wszystkie pliki .jpg zgromadzone w folderze „images” i wygeneruje listę ich miniatur (wystarczy narzucenie max-width, max-height w atrybucie style (CSS)).
2. Kliknięcie na którąś z miniatur oznacza otwarcie strony galery.php z parametrem img=nazwapliku i wówczas poza listą miniatur zostanie wyświetlona pełna wersja grafiki.



## ĆWICZENIE E – SESJE

Sesje umożliwiają zapisanie stanu interakcji klienta z serwerem pomiędzy poszczególnymi wywołaniami. Zapewniają wyższy poziom abstrakcji, niż alternatywne rozwiązania oparte na cookies lub przekazywaniu ukrytych parametrów.

### OBSŁUGA SESJI W PHP:

**A) USTANOWIENIE SESJI:** `session_start();`

Utworzenie nowej sesji lub załadowanie zmiennych dla już istniejącej sesji.

UWAGA! FUNKCJA DOSTĘPU DO SESJI, PODOBNIEM JAK FUNKCJA DODAJĄCA NAGŁÓWEK HTTP - `HEADER('TREŚĆ NAGŁÓWKA')`; MUSZĄ BYĆ W KODZIE STRONY WOŁANE ODPOWIEDNIO WCZEŚNIE, T.J. PRZED KOMPONOWANIEM CIAŁA DOKUMENTU. JEST TO SPOWODOWANE TYM, ŻE FUNKCJE TE TWORZĄ ODPOWIEDNIE NAGŁÓWKI GENEROWANEJ ODPOWIEDZI HTTP ZNAJDUJĄCE SIĘ PRZED CIAŁEM DOKUMENTU.

**B) USTAWIENIE ZMIENNYCH SESJI:** `$_SESSION['nazwaZmiennej']=wartość;`

**C) SPRAWDZANIE OBECNOŚCI ZMIENNEJ SESJI:** `isset($_SESSION['nazwaZmiennej']);`

**D) ODCZYTANIE ZMIENNYCH SESJI:** `$_SESSION['nazwaZmiennej'];`

**E) WYREJESTROWANIE ZMIENNEJ:** `$_SESSION['nazwaZmiennej']=NULL;`

**F) WYREJESTROWANIE WSZYSTKICH ZMIENNYCH:** `session_unset();`

**G) ZNISZCZENIE SESJI:** `session_destroy();`

### ZADANIE:

1. Utwórz stronę  `Dodaj.html` z formularzem zawierającym pole tekstowe i przyciskiem zatwierdzania (`type = submit, value = Dodaj`). Jako stronę docelową formularza określ stronę `podsumuj.php`.
2. Załóżmy, że powyższa strona ma służyć użytkownikowi do dodawania nazw produktów do listy.
3. Strona `podsumuj.php` powinna dodawać nowo przekazany parametr do tablicy przechowywanej w zmiennej sesji (dostępnej poprzez np. `$_SESSION['produkty']` ).  
Uwaga – ponieważ już sama `$_SESSION` jest tablicą, to dostęp do poszczególnych nazw produktów będzie realizowany przez tablicę zagnieżdżoną. Tzn. np. `$_SESSION['produkty'][0]` da nam dostęp do pierwszej z dodanych nazw produktów.
4. Następnie strona `podsumuj.php` powinna wypisywać nazwy wszystkich dodanych produktów.
5. Powinna też udostępniać hiperłącze umożliwiające powrót do strony  `Dodaj.html`. Przechodzimy pomiędzy tymi stronami kilkakrotnie i sprawdzamy, czy mechanizm sesji umożliwił nam zapamiętanie wprowadzonych w poprzednich podejściach nazw.
6. Rozszerz możliwości formularza tak by możliwe było podanie zarówno nazwy produktu jak i ilości. Uwzględnij te dane przy zapisie produktów do sesji i wyświetlaniu listy zakupów.