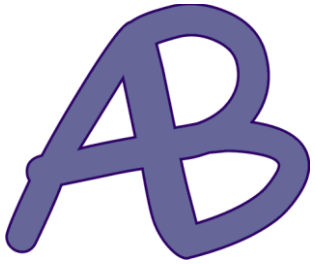


PHP

- funkcje,

dr Artur Bartoszewski
Uniwersytet Radomski

Funkcje

Funkcje

Funkcja, która nie zwraca wartości:

```php
<?php
  function nazwa($parametr01, $parametr02) {
  }
?>
```

Funkcja która zwraca wartość”

```php
<?php
  function nazwa_02($parametr01, $parametr02)
  {
      return 100;
  }
?>
```

Domyślne wartości parametrów:

```php
<?php
  function nazwa($parametr01=0, $parametr02=100) {
  }
?>
```

Przekazywanie parametrów do funkcji

Wartości parametrów przekazywane są do funkcji, a funkcja otrzymuje kopię tych wartości. Zmiany w parametrach wewnątrz funkcji nie wpływają na oryginalne zmienne poza funkcją:

```php
<?php
  function zwieksz_o_jeden($liczba)
  {
    $liczba++;
    echo "wynik= $liczba";
  }
  $x = 5;
  zwieksz_o_jeden($x); //  funkcja wypisze 6, ale $x nadal zawiera 5
?>
```

Przekazywanie parametrów do funkcji

Przekazywanie parametrów przez referencję:

Jeśli chcesz, aby funkcja modyfikowała oryginalne zmienne, możesz przekazać parametr przez referencję, dodając znak "&" przed parametrem.:

```php
<?php
 function zwieksz_o_jeden(&$liczba)
  {
    $liczba++;
    echo "wynik= $liczba";
  }
  $x = 5;
  zwieksz_o_jeden($x); //  funkcja wypisze 6, $x zawiera 6
?>
```

Przekazywanie parametrów do funkcji

Przekazywanie tablicy jako parametru:

- Możesz przekazać tablicę jako parametr do funkcji.
- Nie musimy w żaden specjalny sposób zaznaczać że zmienną którą oczekuje funkcja jest tablica. Może to powodować błędy.
- Tablice przekazywane są zawsze poprzez referencje

```php
<?php
  function suma_tablicy($tablica)
  {
    return array_sum($tablica);
  }

  $liczby = [1, 2, 3, 4, 5];
  echo suma_tablicy($liczby); // Wyświetli 15
?>
```

Przekazywanie parametrów do funkcji

Przekazywanie zmiennej liczby parametrów:

- Możesz przekazywać zmiennej liczby parametrów do funkcji za pomocą specjalnej składni z operatorem ....

```php
<?php
  function srednia(...$liczby)
  {
    $suma = array_sum($liczby);
    $ilosc = count($liczby);
    return $suma / $ilosc;
  }

  echo srednia(1, 2, 3, 4); // Wyświetli 2.5
?>
```

Jak łatwo zauważyć z wszystkich parametrów wysłanych do funkcji budowana jest tablica. Ponieważ w PHP tablice mogą zawierać mieszane wartości parametry nie muszą być tych samych typów należy jednak pamiętać że taką tablicę trzeba będzie zinterpretować wewnątrz funkcji.

Funkcje – zasięg widoczności zmiennych

Zmienne lokalne, czyli zmienne zadeklarowane wewnątrz funkcji,

- są widoczne tylko w jej obszarze (co raczej nikogo nie dziwi)
- po zakończeniu działania funkcji lub bloku, zmienne lokalne przestają istnieć.

```php
<?php
  function nazwa($parametr01, $parametr02) {
      $x = 100;
  }
?>
```

Funkcje – zasięg widoczności zmiennych

Zmienne globalne, Zadeklarowane poza funkcją

- są widoczne poza funkcją lecz nie wewnątrz niej (co może być pewnym zaskoczeniem dla programistów c++)
- są dostępne na całym poziomie skryptu i w obrębie różnych funkcji,
- mogą być używane na całej stronie lub w innych plikach (po ich zaimportowaniu).

```
$x_globalny = 100;
function nazwa($parametr01, $parametr02)
{
    $x = 100;
    echo $x;
    echo $x_globalny; //błąd - zmienna nie jest widoczna
}
```

Funkcje – zasięg widoczności zmiennych

Są dwa sposoby na sięgnięcie do zmiennej globalnej z wnętrza funkcji:

- użycie słowa kluczowego *global*,
- użycie zmiennej super-globalnej *$_GLOBALS['nazwa zmiennej']*

```php
$a = 1;
$b = 2;

function Suma()
{
    global $a, $b;
    $b = $a + $b;
}
```

```php
$a = 1;
$b = 2;

function Suma2()
{
    $_GLOBALS['b'] = $_GLOBALS['a'] + $_GLOBALS['b'];
}
```

Funkcje – zasięg widoczności zmiennych

Tablica $_GLOBALS jest asocjacyjną tablicą, w której nazwa zmiennej jest kluczem, a zawartość zmiennej wartością komórki tablicy.
Zauważmy, że $GLOBALS jest dostępna z każdego miejsca, ponieważ $GLOBALS jest tablicą superglobalną.

```php
<?php
function test()
{
    global $HTTP_POST_VARS;
    echo $HTTP_POST_VARS['name'];
    echo $_POST['name'];
}
?>
```

Większość predefiniowanych zmiennych globalna ale nie "superglobalna" i wymaga 'global', by być dostępnymi w zasięgu lokalnym funkcji

Zmienne **Superglobale** są dostępne z każdego miejsca i nie wymagają 'global'.
Superglobale udostępniono wraz z PHP 4.1.0, a HTTP_POST_VARS jest uważane za przestarzałe.

Zmienne statyczne

Zmienne statyczne są specjalnym rodzajem zmiennych, które zachowują swoją wartość między różnymi wywołaniami funkcji. Są to zmienne, które są inicjowane tylko raz, gdy funkcja, w której się znajdują, jest wywoływana po raz pierwszy, i zachowują swoją wartość pomiędzy kolejnymi wywołaniami tej samej funkcji.

```php
<?php
  function nazwa() {
      static $x = 100; //deklaracja zmiennej statycznej –
                       // ten kod wywoła się tylko przy pierwszym wywołaniu funkcji
      $x++; //operacje na zmiennej statycznej
            //ten kod będzie wykonywany za każdym razem
  }
?>
```

Dopuszczalna składnia:

```php
function funkcja()
{
    static $x = 0;              // prawidłowo
    static $x = 1 + 2;          // błąd  (w rzeczywistości to jest wyrażenie)
    static $x = sqrt(121);      // błąd  (to również jest wyrażenie)
    $x++;
    echo $x;
}
```

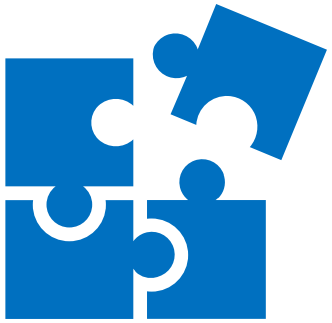
Zmienne statyczne

Przykład” licznik wywołań funkcji

```php
<?php
  function licznik()
  {
    static $licznik = 0; // Deklaracja zmiennej statycznej
    $licznik++; // Inkrementacja zmiennej
    echo "Wartość licznika: $licznik<br>";
  }

  licznik(); // Wartość licznika: 1
  licznik(); // Wartość licznika: 2
  licznik(); // Wartość licznika: 3
?>
```

Przykład do wykonania



Funkcja rekurencyjna – obliczająca silnię

Przykład

```php
<?php
function silnia($n)
{
  if ($n == 0) {
    return 1;
  } else {
    return $n * silnia($n - 1);
  }
}
print("10! = " . silnia(10));
?>
```

Zmienne superglobalne

Zmienne superglobalne

Zmienne superglobalne w języku PHP są specjalnym rodzajem zmiennych, które są dostępne na całym poziomie aplikacji.

Są one wbudowane w język i służą do przechowywania różnych informacji, takich jak dane z formularzy, sesje, nagłówki HTTP i wiele innych.

Zmienne superglobalne rozpoczynają się od znaku dolara i podkreślnika (np. $_GET, $_POST) i są dostępne w każdym miejscu skryptu.

Zmienne superglobalne

$_GET

- Przechowuje dane przesłane z URL za pomocą metody GET, takie jak parametry zapytania.
- Wykorzystywane do odbierania danych od klienta, na przykład formularza HTML.

$_POST

- Przechowuje dane przesłane do serwera za pomocą formularzy HTML z użyciem metody POST.
- Służy do odbierania danych, które nie są widoczne w adresie URL.

$_REQUEST

- Zmienna, która łączy dane przesłane zarówno metodą GET, jak i POST.
- Może być używana do dostępu do danych z dowolnego źródła.

$_GLOBALS

- Przechowuje zmienne globalne dostępne w całym skrypcie.
- Można z niej korzystać do dostępu do zmiennych globalnych w dowolnym miejscu kodu.

Zmienne superglobalne

$_SESSION

- Przechowuje dane sesji użytkownika, umożliwiając zachowanie stanu między różnymi stronami.
- Używana do przechowywania informacji, które powinny być dostępne na wielu stronach podczas sesji użytkownika.

$_COOKIE

- Przechowuje dane przechowywane w ciasteczkach (cookies) na komputerze klienta.
- Wykorzystywane do zapamiętywania informacji na stronie internetowej po odwiedzeniu przez użytkownika.

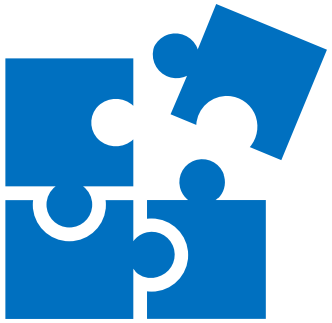
$_SERVER

- Przechowuje informacje o serwerze i żądaniach HTTP.
- Można z niej odczytywać informacje o adresie IP klienta, ścieżce do skryptu, nagłówkach HTTP i innych zmiennych związanych z serwerem.
- jest używane do analizy i kontrolowania żądań HTTP

$_FILES

- Służy do przetwarzania i zarządzania danymi przesłanymi za pomocą plików w formularzach HTML.
- Jest używana do obsługi przesyłania plików przez klientów na serwer, na przykład w przypadku przesyłania obrazów, dokumentów lub innych plików przez formularz na stronie internetowej.

Przykład do wykonania



Odczytaj dane o serwerze przechowywane w tablicy super globalnej

Przykład

Tablica $_SERVER jest asocjacyjną tablicą, w której nazwa zmiennej jest kluczem, a zawartość zmiennej wartością komórki tablicy.

```php
<?php
    foreach ($_SERVER as $klucz => $wartosc)
    echo "<p>$klucz : $wartosc</p>"
?>
```

MIBDIRS : C:/xampp/php/extras/mibs

MYSQL_HOME : \xampp\mysql\bin

OPENSSL_CONF : C:/xampp/apache/bin/openssl.cnf

PHP_PEAR_SYSCONF_DIR : \xampp\php

PHPRC : \xampp\php

TMP : \xampp\tmp

HTTP_HOST : localhost

HTTP_CONNECTION : keep-alive

HTTP_CACHE_CONTROL : max-age=0

HTTP_SEC_CH_UA : "Chromium";v="118", "Microsoft Edge";v="118", "Not=A?Brand";v="99"

HTTP_SEC_CH_UA_MOBILE : ?0

HTTP_SEC_CH_UA_PLATFORM : "Windows"

HTTP_DNT : 1

HTTP_UPGRADE_INSECURE_REQUESTS : 1

HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36 Edg/118.0.2088.69

HTTP_ACCEPT : text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

HTTP_SEC_FETCH_SITE : same-origin

HTTP_SEC_FETCH_MODE : navigate

HTTP_SEC_FETCH_USER : ?1

HTTP_SEC_FETCH_DEST : document

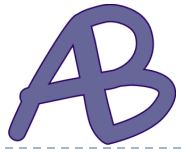
HTTP_REFERER : http://localhost/wyklady/w04/

HTTP_ACCEPT_ENCODING : gzip, deflate, br

HTTP_ACCEPT_LANGUAGE : pl,en;q=0.9

PATH : C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\Git\cmd;C:\Program Files (x86)\MinGW\bin;C:\Program Files\dotnet\;C:\Program Files\Microsoft SQL Server\130\Tools\Binn\;C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Binn\;C:\Program Files\Microsoft SQL Server\150\Tools\Binn\;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;%SYSTEMROOT%\System32\OpenSSH\;C:\Program Files\Microsoft VS Code\bin;C:\Users\artek\.dotnet\tools

SystemRoot : C:\WINDOWS



Literatura

W prezentacji użyto przykładów z książki:
- Żygłowicz Jerzy - PHP - Kompendium wiedzy, Helion

- https://www.php.net