

Wykład: Współczesne zastosowania informatyki II

Autor: dr Aleksander Dawid

email: dawid@us.edu.pl

PHP (*Personal Home Page, Personal Hypertext Preprocessor*)

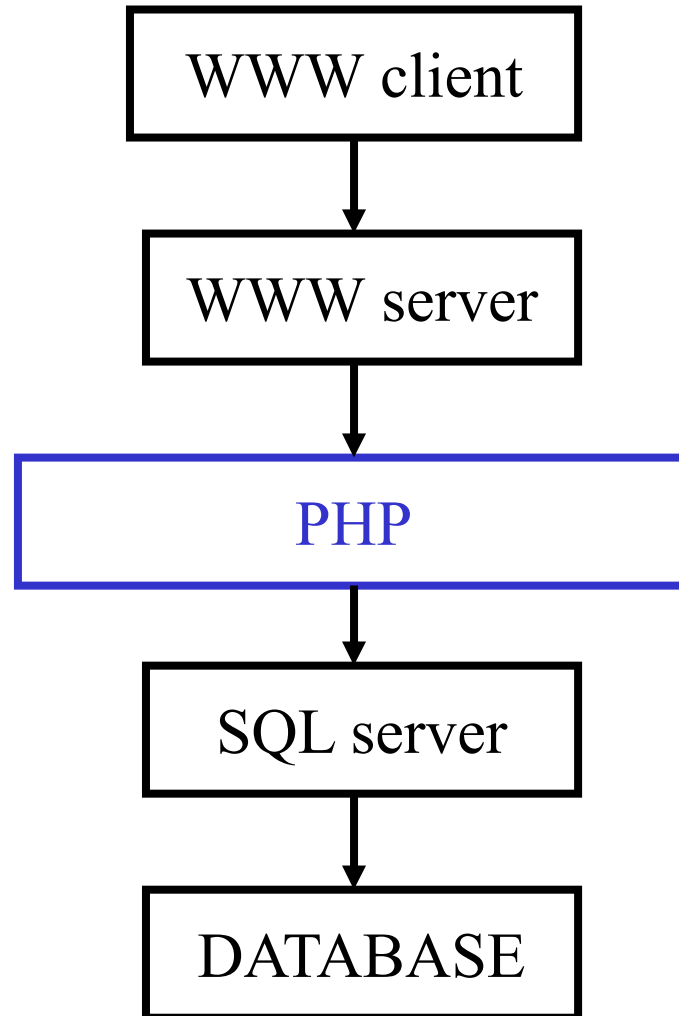
Literatura:

C. Hilton, J. Willis - „PHP3 Internetowe aplikacje bazodanowe”
Helion, 2000 r.

M. Gruber - „SQL”, Helion 1996 r.

<http://www.w3c.org>

Miejsce skryptu w serwisie www



PHP integruje bazy danych z internetem.

Strona <http://www.php.pl>

Różnice między PERL a PHP

- **Zagnieżdzenie w kodzie HTML-a.**
- **Ukierunkowanie na aplikacje bazodanowe.**
- **Wrażliwość na błędy.**

Umieszczenie kodu PHP w kodzie HTML

1. Własny znacznik PHP

```
<H1>Kod HTML</H1>  
<?PHP  
    echo "<P>Skrypt PHP</P>" ;  
?>
```

2. Znaczniki krótkie

```
<H1>Kod HTML</H1>  
<?  
    echo "<P>Skrypt PHP</P>" ;  
?>
```

3. Znaczniki skryptów

```
<H1>Kod HTML</H1>  
<SCRIPT LANGUAGE="PHP">  
    echo "<P>Skrypt PHP</P>" ;  
</SCRIPT>
```

4. Znaczniki ASP

```
<H1>Kod HTML</H1>  
<%  
    echo "<P>Skrypt PHP</P>" ;  
%>
```

Komentarze

```
<?PHP
/* Pierwszy skrypt PHP
Autor: Jan Nowak
*/
    echo "<P>Skrypt PHP</P>"; // pierwsza linia
    # koniec programu
?>
```

Zmienne

Maksymalna długość to 32 znaki. Rozpoczynają się od znaku \$.

Typy zmiennych

- liczby
- łańcuchy znaków.
- tablice.
- obiekty.

Podczas pracy z dokumentami PDF

- Pdfdoc
- Pdfinfo

W celu wymuszenia konwersji stosujemy funkcję **settype ()**

Zmienne

```
$miasto="Warszawa";
```

```
$liczba=3.14;
```

Tablice

Przypisanie sekwencyjne

```
$miasta[]="Warszawa";
```

```
$miasta[]="Jastrzębie";
```

```
$miasta[]="Katowice";
```

Przypisanie indeksom.

```
$miasta[0]="Warszawa";
```

```
$miasta[1]="Jastrzębie";
```

```
$miasta[2]="Katowice";
```

Użycie array.

```
$miasta=array("Warszawa","Jastrzębie","Katowice");
```

```
$miasta=array("Warszawa" => 1000000,
```

```
"Jastrzębie" => 100000,"Katowice" => 400000);
```

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

Inicjalizacja obiektów.

```
class miasto {  
    var $nazwa;  
    var $liczba;  
    function zamknij() {  
        echo "Dostęp do miasta zamknięty";  
    }  
}
```

```
$jas=new miasto;  
$jas->nazwa="Jastrzębie";  
$jas->zamknij();
```

Zmienne globalne w PHP nie są dostępne wewnątrz funkcji.

Używanie zmiennych globalnych w funkcjach.

```
$a=3;  
$b=4;  
function suma()  
{  
    global $a, $b;  
    $b=$a+$b  
}  
suma();  
echo $b;
```

Używanie zmiennych globalnych w funkcjach.

```
$a=3;  
$b=4;  
function suma()  
{  
  $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];  
}  
suma();  
echo $b;
```

Zmienne statyczne

```
function licznik()  
{  
    static $licz=0;  
    $licz++;  
    echo $licz;  
}  
licznik();  
1  
licznik();  
2  
licznik();  
3
```

Zmienne symboliczne.

```
$a="Jan";
```

```
$$a="Nowak";
```

```
echo "$a" . "$Jan"
```

Problem

```
$$a[1]
```

Rozwiązanie

```
${$a[1]}
```

```
${$a}[1]
```

Zmienne zewnętrzne.

GET i POST

Plik HTML

```
<FORM METHOD=POST ACTION="cgi-bin/post.php">  
email: <INPUT TYPE=TEXT NAME="email"><BR>  
Telefon: <INPUT TYPE=TEXT NAME="telefon"><BR>  
</FORM>
```

Plik PHP

```
<?PHP  
    echo("email: $email <BR>");  
    echo("telefon: $telefon");  
?>
```

Cookies

Stanowi fragment nagłówka HTTP

```
Content-type: text/html
```

```
Cookies: name=value&name2=value2
```

Plik PHP

```
<?PHP
```

```
    SetCookie("Identyfikacja", "Jan Nowak");
```

```
?>
```

```
<HTML>
```

```
<BODY>
```

```
...
```

```
</BODY>
```

```
</HTML>
```

Zmienne środowiskowe

Plik PHP

```
<?PHP
    echo "IP: $_SERVER['REMOTE_ADDR']<br>"
?>
```

Operatory takie same jak w C, PERL.

Funkcje

- Funkcje wbudowane
- Funkcje definiowane przez użytkownika.

Definicja funkcji

Plik PHP

```
<?PHP
    function kwadrat( $liczba ){
        return $liczba * $liczba;
    }
    echo kwadrat(4) ;
?>
```

Instrukcje

Podobne do instrukcji w C jak i w PERL
IF, WHILE, DO-WHILE, FOR, SWITCH

Plik PHP

```
<?PHP
    var $a;

    if ($warunek) {
        $a=1;
    }else{
        $a=0
    }

?>
```

Klasy

Podobne do klas w C++

Plik PHP

```
<?PHP
```

```
class nazwa_klasy {
```

```
//Właściwości
```

```
var $zmienna1;
```

```
var $zmienna2;
```

```
//Konstruktor
```

```
function nazwa_klasy() {
```

```
$this->zmienna1=10;
```

```
}
```

```
//Metoda
```

```
function nazwa_metody() {$this->zmienna1++;}
```

```
}//class end
```

```
?>
```

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

Klasy potomne

Plik PHP

```
<?PHP
class nowa_klasa extends nazwa_klasa
{
    //Nowa metoda
    function napis(){
        echo("Zmienna 1: $this->zmienna1);
    }
} //class end
?>
```

Trójargumentowy operator warunkowy

Plik PHP

```
<?PHP
```

```
$jeden = TRUE;
```

```
$dwa = "NIE";
```

```
$trzy = "TAK";
```

```
$jeden ? $dwa : $trzy
```

```
// jeżeli $jeden == FALSE to $dwa
```

```
// w przeciwnym razie $trzy
```

```
?>
```

WYNIK:

```
$jeden = TAK;
```

SQL (*Structured Query Language*)

Język zapytań do bazy danych. IBM lata osiemdziesiąte.
Stosowany w systemach zarządzania bazami danych (DBMS);
Oracle, Paradox, Access itd. .

SQL jest „podjęzykiem danych”. (case insensitive)

Sposoby używania SQL-a

- **Interaktywny lub samodzielny SQL.**
Wprowadzany przez użytkownika w drodze interakcji z programem.
- **Statyczny SQL.**
Zaszyty w kodzie programu wynikowego. Dwie wersje (zanurzony, modułowy).
- **Dynamiczny SQL.**
Kod SQL-a generowany przez aplikację w czasie jej wykonywania

SQL jest językiem nieproceduralnym.

Główne grupy instrukcji SQL-a

- **Język definicji danych (DDL, tworzenie różnych obiektów)**
CREATE SCHEMA, CREATE TABLE, CREATE VIEW
CREATE ASSERTION, CREATE DOMAIN, CREATE INDEX.
- **Język manipulowania danymi(DML)**
SELECT, INSERT, UPDATE, DELETE.
- **Instrukcje sterowania danymi.**
GRANT, REVOKE.
nadawanie/odpieranie uprawnień użytkownikom

CREATE DATABASE

Tworzy bazę danych.

```
CREATE DATABASE pojazd_db;
```

DROP DATABASE

Kasuje bazę danych.

```
DROP DATABASE pojazd_db;
```

SHOW DATABASES

Pokazuje wszystkie bazy danych na serwerze.

```
Show databases;
```

CREATE TABLE

Tworzy tabelę (klasę) na dane.

```
CREATE TABLE auto (  
    NAZWA char(32),  
    ROK_PRODUKCJI date,  
    NUMER int NOT NULL,  
    WAGA float  
);
```

Standardowe typy SQL.

int - liczby całkowite 4 B.

float - zmiennoprzecinkowa.

real - rzeczywista.

smallint - 2 B.

varchar(N) - łańcuch tekstu o stałej długości.

varchar(MAX) - łańcuch tekstu o zmiennej długości.

date - data.

time - czas.

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

CREATE INDEX

Tworzy index dla danej tabeli.

```
CREATE INDEX index_auto ON auto (ROK_PRODUKCJI);
```

DROP INDEX

Kasuje index dla danej tabeli.

```
DROP INDEX index_auto ON auto (ROK_PRODUKCJI);
```

DROP TABLE

Kasuje tabele.

```
DROP TABLE auto;
```

Dodawanie danych do tabeli.

INSERT

```
INSERT INTO auto (NAZWA, ROK_PRODUKCJI, NUMER, WAGA)
VALUES ('Ford fiesta', '6/15/96', 'AAD112321', '1124.34');
```

Rekordy dodawane są na końcu tabeli.

Aktualizacja danych w tabeli.

UPDATE

```
UPDATE auto SET NUMER = 'AAA000000';
```

Cała kolumna NUMER jest zmieniana na nową wartość.

```
UPDATE auto SET NUMER = 'AAA000000'
```

```
WHERE NAZWA = 'Ford fiesta';
```

Modyfikacja rekordu dla którego spełniony jest warunek.

Usuwanie danych z tabeli.

DELETE

```
DELETE FROM auto;
```

Usunięcie wszystkich rekordów z tabeli.

```
DELETE FROM auto WHERE NAZWA='Ford fiesta';
```

Usunięcie rekordu dla którego spełniony jest warunek.

Prezentacja danych z tabeli.

SELECT

```
SELECT * FROM auto;
```

Wyświetlenie wszystkich rekordów z tabeli.

```
SELECT NAZWA, NUMER FROM auto;
```

Wyświetlenie kolumny NAZWA i NUMER.

```
SELECT * FROM auto WHERE NAME='Ford fiesta';
```

Wyświetlenie rekordów dla których spełniony jest warunek.

```
SELECT * FROM auto ORDER BY NAME='Ford fiesta' ;
```

**Wyświetlenie rekordów w porządku określonym kolumną
NAME.**

OPERATORY LOGICZNE(and, or)

```
SELECT * FROM auto
```

```
WHERE NAME='Ford fiesta' and WAGA>'1000' ;
```

**Wyświetlenie rekordów w ze spełnionym pierwszym i drugim
warunkiem.**

Przekierowania zapytań SELECT.

```
SELECT * INTO Temp FROM auto ;
```

Polecenie to w sposób nie jawny tworzy tabelę Temp.

Stosowanie funkcji agregujących.

Count() ; Funkcja zwraca ilość wierszy.

Avg() ; Funkcja zwraca wartość średniej.

Max() ; Funkcja zwraca wartość maksymalną.

Min() ; Funkcja zwraca wartość minimalną.

Sum() ; Funkcja zwraca wartość sumy.

Przykład.

```
SELECT max(WAGA) auto;
```

```
SELECT sum(WAGA) auto;
```

Dostęp do bazy danych MySQL

Inicjalizacja połączenia z serwerem MySQL.

```
int mysql_connect(string nazwa_hosta,  
string nazwa_użytkownika, string hasło);
```

Połączenie to zostanie zerwane po zakończeniu działania skryptu lub po wykonaniu funkcji `mysql_close`.

SUCCESS: identyfikator połączenia.

ERROR: FALSE

Zamknięcie połączenia z serwerem MySQL.

```
int mysql_close()
```

SUCCESS: TRUE.

ERROR: FALSE

Przesłanie zapytania MySQL.

```
int mysql_query(string zapytanie,  
                int id_połączenia)
```

SUCCESS: Identyfikator zbioru wyników.

ERROR: FALSE

Zwraca numer błędu wygenerowanego podczas wykonywania ostatniego polecenia.

```
int mysql_errno(int id_połączenia)
```

SUCCESS: Numer błędu.

ERROR: 0 - nie było błędu.

Zwraca nazwę tabeli na podstawie zbioru wyników

```
string mysql_tablename(int id_wyników,  
                        int i)
```

SUCCESS: nazwa tabeli.

ERROR: NULL

Zwraca ilość wierszy w zbiorze wyników.

```
int mysql_num_rows(int id_wyników)
```

SUCCESS: ilość wierszy w zbiorze wyników.

ERROR: FALSE

Zwraca listę baz danych dostępną na serwerze MySQL.

```
int mysql_list_dbs(void)
```

```
SUCCESS: identyfikator wyników.
```

```
ERROR: FALSE
```

Zwraca zawartość jednej komórki ze zbioru wyników.

```
mixed mysql_result(id wyników,int wiersz  
                    [mixed pole])
```

Wartości zależne od wyniku.

Wybiera bazę danych MySQL .

```
int mysql_select_db(string nazwa_bazy,  
int id_połączenia)
```

SUCCESS: TRUE.

ERROR: FALSE

Przy braku id_połączenia zostanie wybrane ostatnie połączenie do serwera MySQL

Zwraca komunikat błędu wygenerowany podczas wykonywania ostatniego połączenia.

```
string mysql_error(int id_połączenia)
```

SUCCESS: komunikat błędu

ERROR: FALSE

Przykład tworzenie tabel i dopisywanie rekordów.

```
<?PHP
$conn = mysql_connect("localhost");
$SQL='SELECT * FROM osoba;';
$SQLAdd='INSERT INTO osoba
        (IMIE,NAZWISKO,WIEK)
        VALUES ('Jan','Nowak','22');';
$SQLCreate='CREATE TABLE osoba
        (
        IMIE char(32),
        NAZWISKO char(32),
        WIEK smallint
        );';
mysql_select_db("dane_db", $conn);
$wyniki=mysql_query($SQLCreate);
if(mysql_errno())
{echo mysql_errno().": ".mysql_error()."<br>\n";exit;}
```

```
$wyniki=mysql_query($SQLAdd) ;
if(mysql_errno())
{echo mysql_errno() . ":" . mysql_error() . "<br>\n";exit;}

$wyniki=mysql_query($SQL) ;
if(mysql_errno())
{echo mysql_errno() . ":" . mysql_error() . "<br>\n";exit;}

$x=0;
while($x<mysql_num_rows($wyniki))
{
    echo mysql_result($wyniki,$x,"IMIE") . " " .
    mysql_result($wyniki,$x,"NAZWISKO") . "<br>\n";
    $x++;
}
mysql_close($conn) ;
?>
```

Pozostałe funkcje MySQL.

Zwraca ilość wierszy objętych działaniem ostatniego zapytania.

```
int mysql_affected_rows(  
int id_połączenia)
```

Działa w przypadku poleceń DELETE, INSERT, UPDATE.

SUCCESS: ilość wierszy.

ERROR: FALSE

Tworzy bazę danych MySQL.

```
int mysql_create_db(string nazwa_bazy  
int id_połączenia)
```

SUCCESS: TRUE

ERROR: FALSE

```
<?PHP
    $conn = mysql_connect("localhost");
    mysql_create_db("test_db", $conn);
    $wyniki=mysql_list_db();
    $x=0;
    while($x<mysql_num_rows($wyniki)){
        $nazwy_tabel[$x] = mysql_tablename($wyniki,$x);
        echo nazwy_tabel[$x] . "<br>";
        $x++;
    }
    mysql_close();
?>
```

Przesuwa wewnętrzny wskaźnik wiersza.

```
int mysql_data_seek(int id_wyników  
int numer_wiersza)
```

SUCCESS: TRUE

ERROR: FALSE

mysql_fetch_row() zwróci ten wiersz

```
<?PHP
```

```
    $conn = mysql_connect("localhost");  
    mysql_select_db("test_db");  
    $wyniki=mysql_query("SELECT * FROM osoba",$conn);  
    mysql_data_seek($wyniki,2);  
    $wiersz=mysql_fetch_row($wyniki);  
    mysql_close();
```

```
?>
```

Przesyła zapytanie do bazy MySQL.

```
int mysql_db_query(string BazaDanych  
string zapytanie, int id_połączenia)  
SUCCESS: identyfikator zbioru wyników  
ERROR: FALSE
```

```
<?PHP
```

```
    $conn = mysql_connect("localhost");
```

```
    $wyniki=mysql_db_query("test_db", "SELECT * FROM  
    osoba", $conn);
```

```
    mysql_close();
```

```
?>
```

Usuwa bazę danych MySQL.

```
int mysql_drop_db(string BazaDanych  
int id_połączenia)
```

SUCCESS: TRUE

ERROR: FALSE

Pobiera wiersz zwracając go w postaci tablicy.

```
array mysql_fetch_array(int id_wyników[,int $type])  
array mysql_fetch_assoc(int id_wyników)
```

```
$type {MYSQL_NUM, MYSQL_ASSOC, MYSQL_BOTH}
```

SUCCESS: tablica asocjacyjna

ERROR: FALSE

```
<?PHP
    $conn = mysql_connect("localhost");
    mysql_select_db("test_db");
    $wyniki=mysql_query("SELECT * FROM osoba WHERE
                        nazwa='jan' ;", $conn);
    while($student=mysql_fetch_array($wyniki))
        echo $student["IMIE"] . "<br>";
    mysql_close();
?>
```

Pobiera informacje o polu.

```
object mysql_fetch_field(int id_wyników int indeks_pola)
```

SUCCESS: obiekt zawierający informację o polu.

ERROR: FALSE

Właściwości obiektu info.

Table - nazwa tabeli z której pochodzi kolumna

max_length - maksymalna długość kolumny.

Not_null - TRUE gdy kolumna nie może mieć wartości NULL.

Primary_key - TRUE kolumna jest kluczem głównym

unique_key - TRUE wartości kolumny nie mogą się powtarzać.

Multiple_key - TRUE kolumna jest kluczem, którego wartości mogą się powtarzać

Numeric - TRUE dane liczbowe
blob - TRUE kolumna przechowuje dane typu blob.
Type - typ danych kolumny.
Unsigned - TRUE jeśli kolumna przechowuje dane
typu unsigned.
Zerofill - TRUE jeśli pola kolumny wypełnione
są zerami.

Określa wielkość wyników.

`array mysql_fetch_lengths(int id_wyników)`

SUCCESS: tablica długości każdego z pól.

ERROR: FALSE

```
<?PHP
    $conn = mysql_connect("localhost");
    mysql_select_db("test_db");
    $wyniki=mysql_query("SELECT * FROM osoba WHERE
                        nazwa='Jan' ;", $conn);
    $info = mysql_fetch_field($wyniki);

    echo "Nazwa kolumny: " . $info->name . "<br>";
    echo "Nazwa tabeli: " . $info->table . "<br>";

    mysql_close();
?>
```

Zwraca wiersz w postaci obiektu.

```
object mysql_fetch_object(int id_wyników)
```

SUCCESS: obiekt którego właściwości odpowiadają polom pobranego wiersza.

ERROR: FALSE

```
<?PHP
    $conn = mysql_connect("localhost");
    mysql_select_db("test_db");
    $wyniki=mysql_query("SELECT * FROM osoba WHERE
        nazwa='Jan' ;", $conn);
    while($student=mysql_fetch_object($wyniki))
        echo $student->IMIE . "<br>";
    mysql_close();
?>
```

Zwraca wiersz w postaci tablicy o indeksach liczbowych.

```
array mysql_fetch_row(int id_wyników)
```

SUCCESS: tablica zawierająca dane z kolejnego wiersza.

ERROR: FALSE

```
<?PHP
```

```
    $conn = mysql_connect("localhost");
```

```
    mysql_select_db("test_db");
```

```
    $wyniki=mysql_query("SELECT * FROM osoba WHERE  
                        nazwa='Jan' ;", $conn);
```

```
    while($student=mysql_fetch_row($wyniki))
```

```
        echo $student[0] . "<br>";
```

```
    mysql_close();
```

```
?>
```

Podaje nazwę danego pola.

```
string mysql_field_name(int id_wyników,  
int indeks_pola)
```

SUCCESS: zwraca nazwę wskazanego pola.

ERROR: FALSE

Pobiera informacje o ostatnim zapytaniu.

```
int mysql_info(int id_wyników)
```

SUCCESS: ustawia domyślny indeks pola.

ERROR: FALSE

Określa nazwę tabeli wskazanego pola zbioru wyników.

```
string mysql_field_table(int id_wyników,  
int indeks_pola)
```

SUCCESS: zwraca nazwę tabeli, z której pochodzi określone pole zbioru wyników.

ERROR: FALSE

Określa typ pola.

```
string mysql_fetch_type(int id_wyników,  
int indeks_pola)
```

SUCCESS: nazwa określająca typ pola.

ERROR: FALSE

Zwraca flagi podanego pola.

```
string mysql_field_flags(int id_wyników,  
int indeks_pola)
```

SUCCESS: zwraca flagi podanego pola.

ERROR: FALSE

Wartości flag

`not_null` - pole nie może zawierać wartości NULL

`primary_key` - pole jest częścią klucza głównego,

`unique_key` - wartości pola nie mogą się
powtarzać

`multiple_key` - wartości pola mogą się powtarzać

`unsigned` - pole zawiera liczby całkowite bez
znaku.

`zerofill` - wartość pola jest dopełniona zerami.

`binary` - dane binarne.

Określa długość pola.

```
string mysql_field_len(int id_wyników,  
int indeks_pola)
```

SUCCESS: długość określonego pola.

ERROR: FALSE

Zwraca nazwę zestawu znaków.

```
string mysql_client_encoding(int id_wyników)
```

SUCCESS: Zwraca domyślny zestaw znaków.

ERROR: FALSE

Zwalnia pamięć zbioru wyników.

```
int mysql_free_result(int id_wyników)
```

SUCCESS: funkcja zwalnia pamięć zajęta przez zbiór wyników.

ERROR: FALSE

Określa identyfikator wygenerowany podczas ostatniego wywołania polecenia INSERT.

```
int mysql_insert_id(void)
```

SUCCESS: identyfikator.

ERROR: FALSE

Zwraca listę pól zbioru wyników.

```
int mysql_list_fields(string nazwa_bazy,  
string nazwa_tabeli, int id_połączenia)
```

SUCCESS: funkcja pobiera informacje o
podanej tabeli.

ERROR: FALSE

Zwraca listę baz danych dostępnych na serwerze
MySQL.

```
int mysql_list_dbs(void)
```

SUCCESS: identyfikator wyników.

ERROR: FALSE

Zwraca listę tabeli dostępnych w bazie danych.

```
int mysql_list_tables(string nazwa_bazy)
```

SUCCESS: identyfikator wyników.

ERROR: FALSE

Określa ilość pól w zbiorze wyników.

```
int mysql_num_fields(int id_wyników)
```

SUCCESS: ilość pól w podanym zbiorze
wyników.

ERROR: FALSE

Określa trwałe połączenie z serwerem MySQL.

```
int mysql_pconnect(string host, string użytkownik,  
                  string hasło)
```

SUCCESS: identyfikator połączenia.

ERROR: FALSE

Zwraca nazwę tabeli z której pochodzi podane pole.

```
int mysql_tablename(int id_wyników, int i)
```

SUCCESS: łańcuch znaków określający nazwę tabeli.

ERROR: FALSE

Organizacja kodu

REQUIRE

Require (nazwa pliku) - zastępuje zawartością podanego pliku.

Instrukcji tej nie można umieszczać w pętli i wywoływać cyklicznie.

```
require ("header.inc");
```

INCLUDE

include (nazwa pliku) - dołącza zawartość pliku o podanej nazwie.

Instrukcji tą można umieszczać w pętli i wywoływać cyklicznie. Dołączenie zawartości następuje za każdym napotkaniem tej instrukcji.

```
$pliki = array („jeden.inc”, „dwa.inc”, „trzy.inc”);  
for ($i=0; $i<count($pliki); $i++) {  
include ($pliki[$i]);  
}
```

Funkcje obsługi daty oraz czasu.

DATE

Formatuje lokalną datę/czas

```
string date(string format, [int znacznik_czasu])
```

Funkcja zwraca łańcuch zawierający datę oraz czas w odpowiednim formacie.

Ex.

```
echo date ("l,dS F Y h:i:s A");
```

```
"Friday, 1st of January 1999 00:00:00 AM"
```

U - ilość sekund od początku epoki.

Y - rok wyrażony przy użyciu liczby czterocyfrowej

y - rok liczba dwu cyfrowa.

F - miesiąc wyrażony przy użyciu pełnej nazwy.

M. - miesiąc wyrażony przy użyciu trzyliterowego skrótu.

m - miesiąc wyrażony za pomocą liczby.
z - dzień roku wyrażony za pomocą liczby.
d - dzień miesiąca wyrażony za pomocą liczby.
l - dzień tygodnia wyrażony za pomocą pełnej nazwy.
D - dzień tygodnia wyrażony za pomocą trzy literowego skrótu.
w - dzień tygodnia wyrażony za pomocą cyfr.
H - godzina w formacie 24- godzinnym.
h - godzina w formacie 12-godzinnym.
i - minuty wyrażone liczbowo.
s - sekundy wyrażone liczbowo.
A - „AM” lub „PM”
a - „am” lub „pm”
S - angielskie końcówki liczebników porządkowych.

mktime

Obsługa daty i czasu.

```
int mktime(int godziny, int minuty, int sekundy  
,int miesiąc, int dzień, int rok,  
[int czy_zimowy ]
```

Funkcja zwraca UNIX-owy znacznik czasu określony argumentami.

getdate

Obsługa daty i czasu.

```
array getdate(int znacznik_czasu)
```

Funkcja pobiera aktualną datę i czas do tablicy asocjacyjnej.

```
<?  
$data=getdate(mktime());  
echo "Dzisiaj jest " .  
$data[weekday];  
?>
```

”seconds” - sekundy

”minutes” - minuty

”hours” - godziny

”mday” - dzień tygodnia

”wday” - dzień tygodnia wyrażony przy użyciu liczb.

”mon” - miesiąc wyrażony przy użyciu liczb.

”year” - rok wyrażony przy użyciu liczb.

”yday” - dzień roku wyrażony przy użyciu liczby.

”weekday” - dzień tygodnia wyrażony przy użyciu pełnej nazwy.

”month” - miesiąc wyrażony przy użyciu pełnej nazwy.

gmdate

Obsługa daty i czasu.

string gmdate(string format, int znacznik_czasu)

Funkcja zwraca aktualną datą i/lub czas GMT.

getTimeOfDay

Obsługa daty i czasu.

array gettimeofday(void)

Funkcja zwraca tablicę asocjacyjną zawierającą elementy.

sec, usec, minuteswest, dstime.

Jeżeli funkcja zwróci wartość 0 to

znaczy, że system nie dysponuje funkcją gettimeofday.

Funkcje obsługi tablic

array

array array (...)

Funkcja zwraca tablicę, której elementami są argumenty wywołania funkcji.

```
<?PHP
$studenci = array("Jan", "tomasz");
?>
```

array_walk

int array_walk(array tablica, string funkcja)

Wywołuje funkcję i przekazuje do niej każdy element tablicy.

```
<?PHP
function lower($str){
$str=strtolower($str);
}
$a=array("Jan","Adam","Waldek");
array_walk($a,"lower");
do{
    echo current($a) . "<br>";
}while (next($a));
?>
```

current

mixed current(array tablica)

zwraca bieżący element tablicy.

Wywołanie tej funkcji nie modyfikuje położenia wskaźnika.

each

array each(array tablica)

Zwraca kolejną parę klucz - wartość pobraną z tablicy.

```
<?PHP
$studenci=array("Jan","Adam","Waldek");
do{
    echo current($studenci) . "<br>\n";
}while (each($studenci));
?>
```

end

end(array tablica)

Przesuwa wewnętrzny wskaźnik tablicy na jej koniec.

key

`mixed key(array tablica)`

Zwraca klucz (indeks) aktualnego elementu tablicy.

ksort

`int ksort(array tablica)`

Funkcja sortuje zawartość tablicy według wartości kluczy.

```
<?PHP
$id=array("Jan"=>21,"Adam"=>22,"Waldek"=>21);
ksort($id);
do{
    echo key($id) . "ID" . current($id) . "<br>\n";
}while (next($id));
?>
```

list

```
void list(...)
```

Funkcja przypisuje wartości zmiennych tak jak gdyby stanowiły one elementy tablicy.

```
<?PHP
$tab=array(1,2,3);
list($a, $b, $c) = $tab;
echo $a . "<br>\n";
echo $b . "<br>\n";
echo $c . "<br>\n";

list($imie, $telefon)=mysql_fetch_array($w);
?>
```

next

mixed next(array tablica)

Przesuwa wewnętrzny wskaźnik tablicy na następny element.

prev

mixed prev(array tablica)

Cofa wewnętrzny wskaźnik tablicy o jeden.

```
<?PHP
$studenci=array("Jan","Adam","Waldek");
end($studenci);
echo current($studenci) . "<br>\n";
prev($studenci);
echo current($studenci) . "<br>\n";
?>
```

count

int count(mixed zmienna)

Zwraca ilość elementów tablicy.

```
<?PHP
$studenci=array("Jan","Adam","Waldek");
echo "Ilość elementów " . count($studenci);
?>
```

reset

mixed reset(array tablica)

Ustawia wewnętrzny wskaźnik pozycji tablicy na jej pierwszy element.

rsort

void rsort(array tablica)

Sortuje tablicę w kolejności malejącej.

sizeof

```
int sizeof(array tablica)
```

Określa wielkość tablicy, działa dokładnie tak samo jak count.

sort

```
void sort(array tablica)
```

Funkcja sortuje tablicę w kolejności rosnącej.

usort

```
void usort(array tablica function  
funkcja_porównawcza)
```

Funkcja sortuje tablicę przy wykorzystaniu funkcji porównawczej określonej przez użytkownika.

```
<?PHP
function cmp($a,$b) {
    if( $a> $b)
        return -1;
    elseif ($a<$b)
        return 1;
    else
        return 0;
}

$tab=(1,2,3);
usort($tab, cmp);
echo $tab[0] . "<br>\n";
echo $tab[1] . "<br>\n";
echo $tab[2] . "<br>\n";
?>
```

array_push

```
int array_push(array tablica,mixed $wartosc)
```

Wstawianie jednego lub więcej elementów na koniec tablicy

array_pop

```
int array_pop(array tablica)
```

Wstawianie jednego lub więcej elementów na koniec tablicy

```
<?PHP  
$studenci=array("Jan","Adam","Waldek");  
$ostatni=array_pop($studenci);  
echo "Ilość elementów " . count($studenci);  
?>
```

Instrukcja foreach

Wykonuje pętle na elementach tablicy.

```
<?PHP
$studenci=array("Jan","Adam","Waldek");
foreach($studenci as $wartosc){
    echo "Wartosc=$wartosc";
}
?>
```

Funkcje obsługi łańcuchów znaków.

addslashes

string addslashes(string łańcuch_znaków)

Dodaje znaki odwrotnego ukośnika przed określonymi znakami takimi jak (‘) - apostrof, (”) - cudzysłów i (\)-slash

Funkcja ta związana jest z zapytaniami do baz danych.

stripslashes

string stripslashes(string łańcuch_znaków)

Usuwa znaki odwrotnego ukośnika przed określonymi znakami takimi jak (‘) - apostrof, (”) - cudzysłów i (\)-slash

Funkcja ta związana jest z zapytaniami do baz danych.

Funkcje obsługi łańcuchów znaków.

strip_tags

`string strip_tags(string łańcuch_znaków)`

Usuwa znaczniki HTML-a.

```
<?PHP
$text= '<TITLE>TYTUŁ STRONY</TITLE>' ;
echo strip_tags($text) ;
?>
```

chunk_split

```
string chunk_split(string łańcuch, [int co_ile]  
, [string łańcuch_wstawiany])
```

Funkcja wstawia do łańcucha znaków ciąg łańcuch_wstawiany co określoną ilość znaków [int co_ile].

chop

```
string chop(string łańcuch)
```

Funkcja usuwa spacje z końca łańcucha znaków.

Zastosowanie może to znaleźć w bazie danych.

```
<?PHP  
    $word = "ABC   ";  
    echo chop($word);  
?>
```

chr

```
string chr(int kod_ascii);
```

Funkcja zwraca znak po podaniu kodu ascii.

crypt

string crypt(string łańcuch, [string ziarno]);

Funkcja szyfruje ciąg znaków algorytmem DES. Funkcja ta jest nieodwracalna.

```
<?PHP
    $nazwaUzytkownika = "jsmith";
    $haslo = "test";
    echo crypt($haslo, $nazwaUzytkownika);
?>
```

WYNIK

jsOrmWx9.tReo

echo

```
echo(string arg1, [string arg2]);
```

Funkcja wyświetla wszystkie przekazane do niej łańcuchy znaków.

ereg

```
int ereg(string wzor, string przeszukiwany,  
[string odnalezione]);
```

Funkcja przeszukuje łańcuch znaków przy użyciu wyrażeń regularnych.

```
<?PHP
    $URL = "mailto:jan123@us.edu.pl";
    $pat = "@";
    if (!ereg($pat, $URL))
        echo "Wzor nie odnaleziono";
    else
        echo "Wzor odnaleziono";

    $pat = "[0-9]";
    if (ereg($pat, $URL))
        echo "Znaleziono cyfry";

    $pat = "^mailto";
    if (ereg($pat, $URL))
        echo "Znaleziono mailto na początku";
?>
```

ereg_replace

```
int ereg_replace(string wzor, string zamiennik,  
[string poszukiwany]);
```

Funkcja przeszukuje łańcuch znaków do momentu odnalezienia poszukiwanego łańcucha znaków, wtedy zamienia go na wzór.

```
<?PHP  
    $URL = "mailto:jan123@us.edu.pl";  
    $pat = "123";  
    echo ereg_replace($pat, "", $URL);  
?>
```

eregi

```
int eregi(string wzor, string poszukiwany,  
[string odnaleziony]);
```

Funkcja przeszukuje łańcuch znaków nie zwracając uwagi na wielkość znaków.

eregi_replace

```
int eregi_replace(string wzor, string zamiennik,  
[string poszukiwany]);
```

Funkcja przeszukuje łańcuch znaków nie zwracając uwagi na wielkość znaków i zamienia go na wzór określony w zmiennej wzór.

explode

`array explode (string separator,
string łańcuch);`

Funkcja dzieli łańcuch znaków w miejscach wystąpienia określonego łańcucha znaków.

```
<?PHP
    $zdanie = "Mam na rydza chrapke";
    $wyraz = explode(" ", $zdanie);
    for (i=0; i<count($wyraz); i++)
        echo wyraz[i] . "\n";
?>
```

flush

```
int flush(void) ;
```

Funkcja opróżnia bufor wyjściowy. Wywoływana jest zwykle przed operacjami zabierającymi dużo czasu.

```
<?PHP
    echo "Trwa aktualizacja bazy danych.\n";
    flush();
?>
```

htmlspecialchars

```
string htmlspecialchars(string łańcuch) ;
```

Funkcja zmienia określone znaki na symbole HTML.

```
\&'    \&amp;'
\""'"  \&quot;'
\<'    \&lt;'
\>'    \&gt;'
```

ltrim

```
string ltrim(string łańcuch) ;
```

Funkcja usuwa spacje umieszczone na początku łańcucha znaków.

```
<?PHP  
    $tekst="  ABC";  
    echo ltrim($tekst) ;  
?>
```

WYNIK

'ABC'

md5

```
string md5(string łańcuch) ;
```

Funkcja oblicza kod mieszający md5 danego łańcucha znaków.

Używany jest w metodzie autoryzacji DIGEST. Więcej informacji można znaleźć w dokumencie RFC 1321

<http://dns.intergal.com.pl/ftp/rfc/>

```
<?PHP
    $tekst="ABC" ;
    $md5=md5 ($haslo.$text) ;
?>
WYNIK
`534b44a19bf18d20b71ecc4eb77c572f`
```

crypt

```
string crypt(string łańcuch[,string salt]);
```

Asymetryczne kodowanie łańcucha znaków. Algorytm DES

crc32

```
string crc32(string $str);
```

Generator kodu kontrolnego

similar_text

```
int similar_text(string $str1,string $str2);
```

Określa podobieństwo dwóch ciągów znaków

levenshtein

```
int levenshtein(string $st1,string $st2);
```

Odległość levenshteina między dwoma ciągami znaków.

nl2br

```
string nl2br(string łańcuch) ;
```

Funkcja zmienia znaki nowej linii ‘\n’ na znaczniki
 HTML-a.

ord

```
int ord(string łańcuch) ;
```

Funkcja zwraca kod ASCII podanego znaku.

parse_str

```
void parse_str(string łańcuch) ;
```

Funkcja analizuje łańcuch znaków i dzieli go na części.

```
<?PHP
$url="http://localhost/test.php?id=123&name
=jan";
    $url=strtok($url, "?");
    $url=strtok("?");
    parse_str($url);
    echo "ID #: " . $id . "<BR>\n";
    echo "NAME: " . $name . "<BR>\n";
?>
```

parse_url

array parse_url(string łańcuch);

Funkcja analizuje adres URL i zwraca jego elementy w postaci tablicy asocjacyjnej.

Elementy tablicy

scheme - określa protokół.

host - określa adres komputera.

port - określa port.

user - nazwa użytkownika.

pass - hasło.

path - hierarchiczna ścieżka dostępu.

query - łańcuch znaków dotyczący zapytań.

fragment - fragment definiowany punktem
zakotwiczenia #.

print

```
print(string argument);
```

Funkcja wyświetla łańcuch znaków.

printf

```
int printf(string format, [mixed args, ...]);
```

Funkcja wyświetla sformatowany łańcuch znaków. Identyczna funkcja jak w C.

quoted_printable_decode

```
string quoted_printable_decode(string łańcuch);
```

Funkcja konwertująca tekst zakodowany w standardzie quoted-printable do postaci łańcucha składającego się ze znaków 8-bitowych.

```
<?PHP
```

```
    $tekst="w=B1tek";
```

```
    echo quoted_printable_decode($tekst);
```

```
?>
```

```
WYNIK
```

```
`wątek`
```

rawURLdecode

```
string rawurldecode(string łańcuch) ;
```

Funkcja dekoduje łańcuch znaków zakodowany w formacie URL.

Format URL kodowania rozpoczyna się znakiem %, po którym następuje dwucyfrowa liczba heksadecymalna. Więcej w RFC 1738.

```
<?PHP  
    $tekst="jan%40us.edu.pl" ;  
    echo rawurldecode($tekst) ;  
?>  
WYNIK  
`jan@us.edu.pl`
```

rawURLEncode

```
string rawurlencode(string łańcuch) ;
```

Funkcja koduje łańcuch znaków do formatu URL.

similar_text

```
int similar_text(string pierwszy, string drugi,  
[double procent]);
```

Funkcja oblicza podobieństwo dwóch łańcuchów.

split

```
array split(string wyr_reg, string łańcuch,  
[int limit]);
```

Funkcja dzieli łańcuch znaków na części, wykorzystuje przy tym wyrażenia regularne.

```
<?PHP  
    $tekst="jan:11233:/home/jan";  
    $wynik=split(":", $tekst);  
    echo $wynik[1];  
?>
```

sprintf

```
string sprintf(string formatujący,  
[mixed argumenty]);
```

Funkcja zwraca sformatowany łańcuch znaków. Działa tak samo jak funkcja w C.

strchr

```
string strchr(string łańcuch, string sz_znak);
```

Funkcja odnajduje pierwsze wystąpienie znaku.

Podobna do strstr().

```
<?PHP  
    $tekst="Mam pytanie o php";  
    echo strchr($tekst,"php");  
?>
```

strcmp

```
int strcmp(string łańcuch1, string łańcuch2);
```

Funkcja binarnie porównuje łańcuch znaków.

Jest to funkcja z ANSI C.

strcspn

```
int strcspn(string łańcuch1, string znaki);
```

Funkcja określa długość początkowego fragmentu łańcucha znaków nie pasującego do podanej maski.

```
<?PHP
```

```
    $url="http://www.php123.com/test.php?id=21";
```

```
    echo "Zapytanie rozpoczyna się od" .
```

```
    strcspn($url,"?") . "znaku.<BR>";
```

```
?>
```

stripslashes

```
string stripslashes(string łańcuch) ;
```

Funkcja usuwa znaki odwrotnego ukośnika.

strlen

```
int strlen(string łańcuch) ;
```

Funkcja zwraca długość łańcucha znaków.

strrpos

```
int strrpos(string łańcuch, string znak,  
int [offset]) ;
```

Funkcja określa położenie ostatniego wystąpienia znaku.

strpos

```
int strpos(string łańcuch, string znak,  
int [offset]);
```

Funkcja określa położenie pierwszego wystąpienia znaku.

strrchr

```
int strrchr(string łańcuch, string poszukiwany);
```

Funkcja wyświetla znaki od momentu ostatniego wystąpienia znaku poszukiwanego.

```
<?PHP  
    $tekst="Mam pytanie o program";  
    echo strrchr($tekst,"p");  
?>  
WYNIK  
program
```

strrev

```
int strrev(string łańcuch);
```

Funkcja zapisuje łańcuch znaków w odwrotnej kolejności.

strtok

```
int strtok(string łańcuch, string separator);
```

Dzieli łańcuch znaków na leksemy.

```
<?PHP
    $str="Mam pytanie o program";
    $slowo=strtok($str," ");
    while ($slowo){
        echo $slowo; $slowo=strtok(" ");
    }
?>
```

WYNIK

Mam pytanie o program

strtolower

```
int strtolower(string łańcuch) ;
```

Zamienia wszystkie litery na małe.

strtoupper

```
int strtoupper(string łańcuch) ;
```

Zamienia wszystkie litery na duże.

substr

```
int strtoupper(string łańcuch, int początek,  
[int długość]) ;
```

Zwraca fragment łańcucha znaków.

Funkcje obsługi sieci

checkdnsrr

```
int checkdnsrr(string host, [string typ]);
```

Sprawdza rekordy DNS różnych typów na podstawie podanego adresu IP lub nazwy hosta.

```
<?php  
echo "Adres IP - " .  
checkdnsrr("www.us.edu.pl", "A") . "<br>;  
echo "Serwer poczty " .  
checkdnsrr("us.edu.pl", "MX") . "<br>;  
?>
```

Funkcje obsługi sieci

Typ

Opis

A – adres IP

NS – serwer nazw

MX – serwer wymiany poczty, preferencje

SOA – wpis administracyjny (domena początkowa, adres domeny zarządcy, numer administracyjny, czas odświeżania, ponownej próby, wygaśnięcia ważności, minimalny czas TTL. RFC 883

NULL – pusty rekord zasobów.

CNAME – pseudonim domeny.

RP – osoba odpowiedzialna za domenę.

PTR – wskaźnik nazwy domeny.

HINFO – informacje o hoście (typ procesora i system operacyjny) .

Gethostbyname

string gethostbyname (string nazwy_hosta) ;

Zwraca adres IP na podstawie jego nazwy.

```
<?  
echo gethostbyname ("localhost") . "<BR>" ;  
echo gethostbyname ("uranos.cto.us.edu.pl") . "<BR>" ;  
echo gethostbyname ("www.nie_ma.pl") . "<BR>" ;  
?>
```

WYNIK:

127.0.0.1

155.158.102.7

www.nie_ma.pl

Gethostbyaddr

```
string gethostbyaddr(string adresIP);
```

Określa nazwę hosta na podstawie jego adresu IP.

```
<?  
echo gethostbyaddr("127.0.0.1") . "<BR>";  
echo gethostbyaddr("155.158.102.7") . "<BR>";  
echo gethostbyaddr("2.0.0.0") . "<BR>";  
?>
```

WYNIK:

localhost

uranos.cto.us.edu.pl

2.0.0.0

fsockopen

```
int fsockopen(string nazwy_hosta, int port,  
[int blad_nr], [string blad_str], [double czas]);
```

Otwiera połączenie przez gniazdko TCP.

```
<?php  
$web=fsockopen("www.us.edu.pl", 80);  
fputs($web, GET / HTTP/1.1\r\n\r\n);  
while(!feof($web)) {  
    $str=fgets($web, 1024);  
    if(!strcmp($str, "\r\n"));  
        break;  
    echo $str . "<br>";  
}  
fclose($web);  
?>
```

mail

```
bool mail(string odbiorca, string temat,  
string wiadomosc, [string dodatkowe_naglowki]);
```

Wysyła wiadomość poczty elektronicznej.

```
<?php  
$tekst="Temat\nNowa linia";  
mail("dawid@us.edu.pl","Temat listu",$text);  
mail("dawid@us.edu.pl","Temat listu",$text,  
"X=Priority: 1");  
mail("dawid@us.edu.pl","Temat listu",$text,  
"From: Test<test@php.com>\nX=Priority: 1");  
);  
?>
```

Funkcje systemowe

passthru

```
string passthru(string polecenie, [int  
zmienna_wynikowa]);
```

Wykonuje podany program i wyświetla zwrócone przez niego wyniki.

```
<?php  
header("Content-type: image/gif");  
passthru("cat /tmp/test.gif");  
?>
```

Funkcje systemowe

usleep

```
void usleep(int mikro_sekund) ;
```

Wstrzymuje wykonywanie skryptu na podaną ilość mikrosekund.

```
<?php  
echo date("h:i:s") . "<br>" ;  
usleep(5000000) ;  
echo date("h:i:s") . "<br>" ;  
?>
```

Funkcje systemowe

uniqid

```
void uniqid(string prefiks) ;
```

Generuje unikalny identyfikator.

```
<?php  
echo uniqid("") . "<br>" ;  
echo uniqid("uni") . "<br>" ;  
?>
```

WYNIK:

38c9276deafa5

uni38c9276deafa5

Funkcje systemowe

phpVersion

```
string phpversion(void) ;
```

Zwraca aktualną wersję PHP.

```
<?php  
echo "Wersja PHP: " . phpversion() ;  
?>
```

WYNIK:

```
Wersja PHP: 4.01
```

Funkcje systemowe

phpInfo

```
string phpinfo(void) ;
```

Wyświetla informacje dotyczące PHP.

```
<?php  
echo phpinfo() ;  
?>
```

WYNIK:

Informacje w postaci tabelki

Funkcje obsługi plików

basename

```
string basename (string path) ;
```

Ekstrakcja nazwy pliku z pełnej ścieżki.

```
<?php  
$str=basename ("C:\Apache\htdocs\test.htm") ;  
echo $str;  
?>
```

WYNIK:

```
test.htm
```

Funkcje obsługi plików

dirname

```
string dirname(string path) ;
```

Zwraca ścieżkę katalogu.

```
<?php  
$str=dirname("C:\Apache\htdocs\test.htm") ;  
echo $str ;  
?>
```

WYNIK:

```
C:\Apache\htdocs
```

Funkcje obsługi plików

chdir

```
int chdir(string katalog);
```

Zmienia katalog. (TRUE/FALSE).

```
Lista plików w katalogo Marek
<?php
chdir("/marek");
system("dir");
?>
```

Funkcje obsługi plików

dir

`new dir(string katalog) ;`

Zwraca klasę katalogu.

```
<?php
$dir=new("/marek");
while($plik=$dir->read()){
    echo $plik . "<br>";
}
?>
```

dir

`handle` - uchwyt do katalogu.

`path` - pełna ścieżka dostępu do katalogu.

`read()` - odczyt katalogu.

`rewind()` - przewijanie do początku.

`close()` - zamykanie dostępu do katalogu.

Funkcje obsługi plików

copy

```
int copy(string plik_źródłowy, string  
plik_docelowy);
```

Kopiowanie plików.

```
<?php  
if (!copy("C:\www\index.htm", "C:\apache\htdocs  
\index.html")) {  
echo("Kopiowanie nie udane");  
}  
?>
```

Funkcje obsługi plików

fopen

```
int fopen(int nazwa_pliku, string tryb) ;
```

Otwiera plik lub dokument o podanym adresie URL.

tryby

r - tryb do odczytu.

r+ - tryb do odczytu i zapisu.

w - tryb do zapisu.

w+ - tryb do zapisu i odczytu.

a - tryb do zapisu. Wskaźnik na końcu pliku.

a+ - tryb do odczytu i zapisu. Wskaźnik na końcu pliku.

@fopen - komunikaty o błędach.

Funkcje obsługi plików

fopen

```
<?php
$fp=fopen("http://www.example.com/","r");
do{
$linia = fgets($fp,1024);
echo $linia;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fclose

```
int fclose(int wskaźnik_pliku);
```

Zamyka otworzony plik
fopen(),fsockopen().

```
<?php  
$fp=fopen("E:\docs\opis.txt","r");  
echo fgets($fp,1024);  
fclose($fp);  
?>
```

Funkcje obsługi plików

feof

```
int feof(int wskaźnik_pliku) ;
```

Sprawdza przekroczenie końca pliku za pomocą wskaźnika.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
do{
$sekcja = fgets($fp,1024);
echo $sekcja;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fgetc

```
int fgetc(int wskaźnik_pliku);
```

Pobiera znak z pliku określonego przy użyciu wskaźnika.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
do{
$sekcja = fgetc($fp);
echo $sekcja;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fgets

`int fgets(int wskaźnik_pliku, int długość);`

Odczytuje linię z pliku określonego przy użyciu wskaźnika.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
do{
$sekcja = fgets($fp,1024);
echo $sekcja;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fread

```
int fread(int wskaźnik_pliku, int długość) ;
```

Odczytuje linię z pliku określonego przy użyciu wskaźnika.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
do{
$sekcja = fread($fp,1024);
echo $sekcja;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fgetss

```
int fgetss(int wskaźnik_pliku, int długość) ;
```

Odczytuje linię z pliku określonego przy użyciu wskaźnika i dodatkowo likwiduje wszystkie znaczniki.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
do{
$linia = fgetss($fp,1024);
echo $linia;
}while(!feof($fp));
fclose($fp);
?>
```

Funkcje obsługi plików

fputs

```
int fputs(int wskaźnik_pliku, string  
łańcuch, int długość);
```

Zapisuje tekst do pliku.

```
<?php  
$fp=fopen("E:\docs\opis.txt","w+");  
fputs($fp,"Tekst w pliku\n");  
rewind($fp);  
$linia = fgets($fp,1024);  
echo $linia;  
fclose($fp);  
?>
```

Funkcje obsługi plików

fwrite

```
int fwrite(int wskaźnik_pliku, string  
łańcuch, int długość);
```

Zapisuje do pliku łańcuch znaków(zapis binarny).

```
<?php  
$fp=fopen("E:\docs\opis.txt","w+");  
fwrite($fp,"Tekst w pliku\n");  
rewind($fp);  
$linia = fgets($fp,1024);  
echo $linia;  
fclose($fp);  
?>
```

Funkcje obsługi plików

fseek

```
int fseek(int wskaźnik_pliku, int  
przesunięcie);
```

Przesuwa w pliku wskaźnik położenia.

```
<?php  
$fp=fopen("E:\docs\opis.txt","r");  
fseek($fp,3);  
$znak = fgetc($fp);  
echo $znak;  
fclose($fp);  
?>
```

Funkcje obsługi plików

ftell

```
int ftell(int wskaźnik_pliku);
```

Zwraca wartość wskaźnika położenia w pliku.

```
<?php
$fp=fopen("E:\docs\opis.txt","r");
echo ftell($fp) . "<br>";
$linia = fgets($fp,1024);
echo $linia;
echo ftell($fp) . "<br>";
fclose($fp);
?>
```

Funkcje obsługi plików

file

```
array file(string nazwa_pliku);
```

Odczytuje cały plik i umieszcza jego zawartość w tablicy.

```
<?php
$tab=file("E:\docs\opis.txt","r");
$x=0;
while($tab[$x]){
    echo $tab[$x++] . "<br>";
}
?>
```

Funkcje obsługi plików

file_exist

```
int file_exist(string nazwa_pliku);
```

Sprawdza czy plik istnieje.

```
<?php
if(file_exist("C:\autoexec.bat")) {
    echo "Jest autoexec" . "<br>";
}
?>
```

Funkcje obsługi plików

filesize

```
int filesize(string nazwa_pliku);
```

Zwraca wielkość pliku.

```
<?php
$rozmiar=filesize("C:\autoexec.bat");
    echo "Rozmiar pliku autoexec.bat: " .
$rozmiar;
}
?>
```

Ładowanie plików na serwer

`$_FILES`

Dwuwymiarowa tablica asocjacyjna.

```
$_FILES['id']['tmp_name'];
```

Przechowuje nazwę pliku tymczasowego na serwerze.

```
$_FILES['id']['name'];
```

Przechowuje oryginalną nazwę pliku.

```
$_FILES['id']['size'];
```

Przechowuje rozmiar pliku.

```
$_FILES['id']['type'];
```

Przechowuje typ pliku.

```
$_FILES['id']['error'];
```

Przechowuje informacje o błędach.

Ładowanie plików na serwer

move_uploaded_file

```
bool move_uploaded_file(string  
$nazwa_tmp, string $nazwa_dst);
```

Kopiuje załadowany plik z katalogu tymczasowego do podanej lokalizacji w systemie operacyjnym

```
<?php  
move_uploaded_file($_FILES['id'][tmp_name], "C:\nazwa.p  
liku") or die ("Bład transmisji");  
?>
```

Ładowanie plików na serwer

KOD HTML FORMULARZ

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<FORM ACTION=upload.php ENCTYPE=multipart/form-data
METHOD=post>
<INPUT TYPE=hidden NAME=MAX_FILE_SIZE VALUE=2000000>
<INPUT TYPE=file name=plik id=plik>
<INPUT TYPE=submit VALUE=wysylaj>
</BODY>
</HTML>
```

Ładowanie plików do katalogów

upload.php

```
<?PHP
$PATH="C:\obrazki";
$DST=$PATH.$_FILES['plik']['name'];
move_uploaded_file($_FILES['plik']['tmp_name'],$DST)
or die(„Brak pliku“.$_FILES['plik']['tmp_name']);
?>
```

Ładowanie plików do bazy danych

budujbaze.php

```
<?PHP
$db_name = "NazwaBazy";
$upload = "CREATE TABLE upload (
id INT NOT NULL AUTO_INCREMENT,
nazwa VARCHAR(30) NOT NULL,
typ VARCHAR(30) NOT NULL,
rozmiar INT NOT NULL,
zawartosc MEDIUMBLOB NOT NULL,
PRIMARY KEY(id)
)";
$conn = mysql_connect("localhost");
@mysql_select_db($db_name, $conn);
$status = mysql_query($upload) or
die("UPLOAD:" . mysql_errno() . ":" . mysql_error() . "<br>\n"
);
mysql_close($conn);
?>
```

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

Ładowanie plików do bazy danych

upload.php

```
<?PHP
$fileName = $_FILES['plik']['name'];
$tmpName  = $_FILES['plik']['tmp_name'];
$fileSize = $_FILES['plik']['size'];
$fileType = $_FILES['plik']['type'];

$fp      = fopen($tmpName, 'r');
$content = fread($fp, filesize($tmpName));
$content = addslashes($content);
fclose($fp);

if(!get_magic_quotes_gpc())
{
    $fileName = addslashes($fileName);
}
```

Ładowanie plików do bazy danych

upload.php

```
$db_name="NazwaBazy";

$conn = mysql_connect("localhost");
$query = "INSERT INTO upload (nazwa, rozmiar, typ,
zawartosc ) ".
"VALUES ('$fileName', '$fileSize', '$fileType',
'$content')";

@mysql_select_db($db_name, $conn);
$status=mysql_query($query) or
die("UPLOAD:".mysql_errno().":".mysql_error()."<br>\n"
);
mysql_close($conn);

echo "<br>File $fileName uploaded<br>";
?>
```

Pobieranie plików z bazy danych

download.php

```
<?PHP
$db_name="NazwaBazy"; $id = $_GET['id'];
$query = "SELECT nazwa, typ, rozmiar, zawartosc " .
        "FROM upload WHERE id = '$id'";
$conn = mysql_connect("localhost");
@mysql_select_db($db_name, $conn);
$result = mysql_query($query) or die('Blad');
list($nazwa, $typ, $rozmiar, $zawartosc) =
mysql_fetch_array($result);
header("Content-length: $rozmiar");
header("Content-type: $typ");
header("Content-Disposition: attachment;
filename=$nazwa");
echo $zawartosc;
mysql_close($conn);
exit;
?>
```

Moduł dodatkowy - grafika w PHP

Instalacja modułów dodatkowych.

Plik php.ini

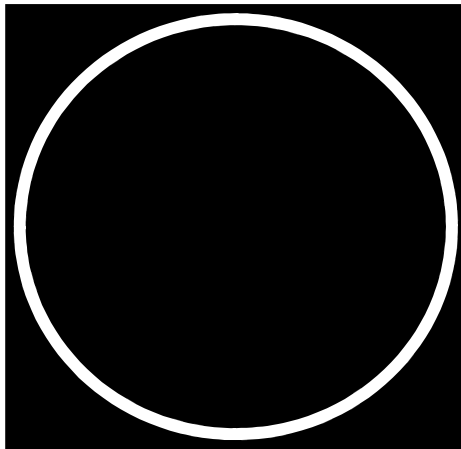
```
extensions=php_gd.dll
```

Rozszerzenia graficzne pozwalają generować obrazy w przeglądarkach sieciowych. Aktualnie dostępne formaty to png i jpg.

ZASTOSOWANIE

- wykresy analizujące jakąś statystykę .
- interface graficzny
- autoryzacja.

```
<?PHP //Rysowanie okręgu.  
Header("Content-type: image/png");  
$img = ImageCreate(250, 250);  
$czarny = ImageColorAllocate($img, 0, 0, 0);  
$biały = ImageColorAllocate($img, 255, 255, 255);  
ImageFill($img, 0, 0, $czarny);  
ImageArc($img, 125, 125, 250, 250, 0, 360, $biały);  
ImagePNG($img);  
ImageDestroy($img);  
?>
```



Header

```
int header(string naglowek);
```

Przesyła nagłówek HTTP.

imageCreate

```
int imageCreate(int szerokość,int wysokość);
```

Funkcja tworzy pusty obrazek i tworzy dla niego identyfikator.

imageColorAllocate

```
int imageColorAllocate(int id,  
int R, int G, int B);
```

Funkcja zwraca identyfikator koloru utworzony przy użyciu trzech liczb (RGB).

imageFill

```
int imageFill(int id_obrazka,int x, int y,  
int kolor);
```

Funkcja wypełnia obrazek podanym kolorem rozpoczynając w punkcie x, y.

imageArc

```
int imageArc(int id_obrazka, int cx, int cy,  
int w, int h, int s, int e, int kolor);
```

Rysuje fragment elipsy.

imageGIF, imagePNG, imageJPG

```
int imagePNG(int id_obrazka,  
string nazwa_pliku);
```

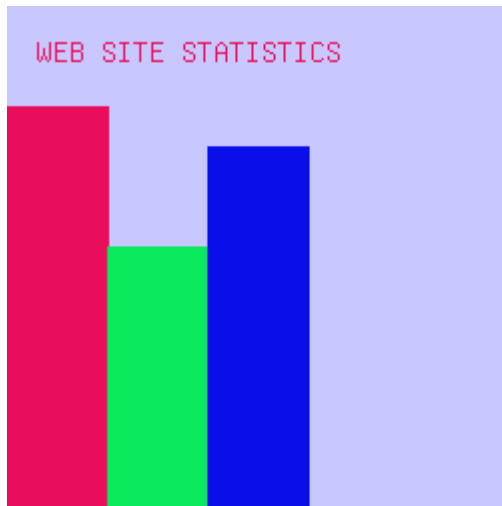
Wyświetla obrazek w przeglądarce lub zapisuje go do pliku o odpowiednim formacie.

imageDestroy

```
int imageDestroy(int id_obrazka);
```

Funkcja zwalnia pamięć używaną do przechowywania danych obrazka.

```
<?php
$title="WEB SITE STATISTICS";
header ("Content-type: image/png");
$im = ImageCreate (250, 250)
$background_color =
ImageColorAllocate ($im, 200, 200, 255);
$czar = ImageColorAllocate ($im, 233, 14, 91);
$ziel = ImageColorAllocate ($im, 10, 233, 91);
$nieb = ImageColorAllocate ($im, 10, 14, 233);
ImageString ($im, 4, 15, 15, $title, $czar);
ImageFilledRectangle ($im, 0, 50, 50, 250, $czar);
ImageFilledRectangle ($im, 50, 120, 100, 250, $ziel);
ImageFilledRectangle ($im, 100, 70, 150, 250, $nieb);
ImagePng ($im);
ImageDestroy ($im);
?>
```



Obrazek wygenerowany przez ostatni skrypt.

Zmienna **background_color** ustawia kolor tła.

imageString

```
int imageString(int id_obrazka, int czcionka  
int x, int y, string łańcuch, int kolor);
```

Funkcja wyświetla łańcuch znaków zaczynając od współrzędnych x, y.

imageFilledRectangle

```
int imageFilledRectangle(int id_obrazka,  
int x1, int y1, int x2, int y2, int kolor);
```

Funkcja rysuje prostokąt i wypełnia go określonym kolorem.

```
<?php
#-----
# LICZNIK GRAFICZNY
#-----
header ("Content-type: image/png");
$str="$number";
$num_width=strlen($str);
#-----
# INICJALIZACJA KATALOGU I PLIKOW
#-----
for ($i=0;$i<10;$i++)
    {
        $nums[$i]=$katalog . "/" . $i . ".png";
    }

$rozmiar=Getimagesize ($nums [0] );
$szer=$rozmiar [0] ;
$wys=$rozmiar [1] ;

$final = ImageCreate ($szer*$num_width, $wys);
```

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

```
#-----  
# Skladanie obrazka w jedna calosc  
#-----  
for($i=0;$i<$num_width;$i++)  
{  
    $im=ImageCreateFromPng($nums[$str[$i]]);  
    Imagecopyresized($final,$im,  
    $i*$szer,0,0,0,$szer,$wys,$szer,$wys);  
    ImageDestroy($im);  
}  
  
$strans=ImageColorAt($final,0,0);  
ImageColorTransparent($final,$strans);  
  
ImagePng($final);  
ImageDestroy($final);
```

?>

Przykład: 1875

Komenda wstawiająca licznik na stronę

```
<IMG SRC=licznik.php?katalog=cyf1&number=$count>
```

Gdzie;

\$count jest zmienną w której znajduje się aktualny stan licznika.

katalog, **number** są parametrami wywołania licznik.php.

imageCreateFromPNG

```
int imageCreateFromPNG(string nazwa_pliku);
```

Funkcja tworzy nowy obrazek na podstawie zawartości podanego pliku lub na podstawie adresu URL.

Getimagesize

```
array getimagesize(string nazwa pliku,  
[array informacje]);
```

Określa wielkość obrazków GIF, JPEG, PNG.

Zwracana tablica

`$tab[0]` - szerokość obrazka w pixelach

`$tab[1]` - wysokość obrazka w pixelach

`$tab[2]` - flaga określająca typ obrazka (1-GIF,2-JPG, 3-PNG)

`$tab[3]` - "height=Y width=X"

imageCopyResized

```
int imageCopyResized(int id_dcl, int id_zrd,  
int dclX, int dclY, int zrdX, int zrdY,  
int dclW, int dclH, int zrdW, int zrdH);
```

Funkcja kopiuje prostokątny obszar jednego obrazka i umieszcza go na drugim obrazku.

PARAMETRY:

id_dcl - identyfikator docelowego obrazka.

id_zrd - identyfikator źródłowego obrazka.

dclX, dclY - współrzędne docelowe.

zrdX, zrdY - współrzędne źródłowe.

dclW, dclH - docelowy rozmiar obrazka.

zrdW, zrdH - źródłowy rozmiar obrazka.

imageColorAt

```
int imageColorAt(int id, int X, int Y);
```

Funkcja zwraca indeks koloru podanego piksela obrazka.

imageColorTransparent

```
int imageColortransparent(int id, int kolor);
```

Funkcja ustawia podany kolor jako przezroczysty.

Funkcje rysujące obiekty

imageLine

```
int imageLine(int id, int x1, int y1,  
int x2, int y2, int kolor);
```

Rysuje linię.

imageDashedLine

```
int imageDashedLine(int id, int x1, int y1,  
int x2, int y2, int kolor);
```

Rysuje linię przerywaną.

imageFilledPolygon

```
int imageFilledPolygon(int id, array punkty,  
int ilość_punktów, int kolor);
```

Rysuje wypełniony wielokąt.

imageFilledPolygon

```
int imageFilledPolygon(int id, array punkty,  
int ilość_punktów, int kolor);
```

Rysuje wypełniony wielokąt.

```
<?php  
header ("Content-type: image/png");  
$im = ImageCreate (250, 250)  
$background_color =  
ImageColorAllocate ($im, 200, 200, 255);  
$czer = ImageColorAllocate ($im, 233, 14, 91);  
$a[0]=10; $a[1]=10;  
$a[2]=240; $a[3]=10;  
$a[4]=240; $a[5]=240;  
$a[6]=10; $a[7]=240;  
ImageFilledPolygon($im,$a,4,$czer);  
ImagePng ($im);  
ImageDestroy($im);  
?>
```

imagePolygon

```
int imagePolygon(int id, array punkty,  
int ilość_punktów, int kolor);
```

Rysuje wielokąt.

imageRectangle

```
int imageRectangle(int id, int x1,int y1,  
int x2,int y2, int kolor);
```

Rysuje prostokąt.

imageSetPixel

```
int imageSetPixel(int id,  
int x,int y, int kolor);
```

Rysuje pojedynczy pixel.

imageSetPixel

```
int imageSetPixel(int id,  
int x,int y, int kolor);
```

Rysuje pojedynczy pixel.

Funkcje operujące na kolorach

imageColorClosest

```
int imageColorClosest(int id,  
int R,int G, int B);
```

Zwraca indeks najbliższego koloru.

imageColorExact

```
int imageColorExact(int id,  
int R,int G, int B);
```

Zwraca indeks podanego koloru.

imageColorResolve

```
int imageColorResolve(int id,  
int B,int G, int B);
```

Zwraca indeks podanego koloru lub najbliższego podanego koloru.

imageColorSet

```
int imageColorSet(int id, int indeks  
int B,int G, int B);
```

Określa kolor podanego indeksu palety.

imageColorsForIndeks

```
array imageColorsForIndeks(int id, int indeks);
```

funkcja zwraca tablicę asocjacyjną o kluczach
"red", "green" oraz "blue".

imageColorsTotal

```
int imageColorsTotal(int id_obrazka);
```

Zwraca ilość kolorów znajdujących się w paletce obrazka.

imageInterlace

```
int imageInterlace(int id_obrazka  
int przeplot);
```

Włącza lub wyłącza przeplot.

Sesje

- Identyfikacja użytkownika
- Identyfikacja obszaru serwisu www
- Mechanizm blokowania stron

Użytkownik otrzymuje unikalne id

Obsługa sesji dodana została w PHP 4.0.0

```
Tablica superglobalna  
$_SESSION
```

Rejestracja sesji

```
<?PHP
session_start();
If(!isset($_SESSION['id'])) {
$_SESSION['id']=0;
}else{
$_SESSION['id']++;
}
?>
```

Wyrejestrowanie sesji

```
<?PHP
session_start();
unset($_SESSION['id']);
?>
```

bool session_start(void)

Funkcja tworzy sesje lub wznawia bieżącą na podstawie aktualnego identyfikatora sesji.

Przekaz identyfikatora

Metody POST,GET

Mechanizm ciasteczek (Cookies)

```
<?PHP
session_start();
$_SESSION['kolor']='czerwony';
//Metoda 1
echo '<a href="podstrona.php">Podstrona</a>';
//Metoda 2
echo "<a href=podstrona.php?\" .SID.\">Podstrona</a>";
?>
```

Sesje

string session_name([string \$nazwa])

Funkcja pobiera i ustawia nazwę bieżącej sesji

```
<?PHP
session_start();
$stara_nazwa=session_name("okazja");
echo "<a href=podstrona.php?" .SID.">Podstrona</a>";
?>
```

```
$stara_nazwa=session_name("(okazja)!");
```

Sesje

string session_id([string \$id])

Funkcja pobiera i ustawia identyfikator bieżącej sesji.

Informacje tą zawiera także zmienna SID.

```
<?PHP
session_start();
$sesja=session_id();
echo "<a href=podstrona.php?\" .
$sesja.\">Podstrona</a>\";
?>
```

Sesje

void session_write_close(void)

Funkcja kończy bieżącą sesję i zapisuje dane sesji.

```
<?PHP
session_start();
...
Operacje na sesji
...
session_write_close();
?>
```

Sesje

string session_encode(void)

Funkcja zwraca ciąg tekstowy z zakodowanymi danymi sesji

bool session_decode([string \$dane])

Funkcja dekoduje dane sesji ze stringu.

PHP5 programowanie obiektowe

Klasy

```
<?PHP
class Budynek
{
    public $nazwa = 'Dowolna wartosc' ;

    public function Wypisz() {
        echo $this->nazwa;
    }
}

$Dom=new Budynek() ;
$Dom->Wypisz() ;
?>
```

PHP5 programowanie obiektowe

Prosta klasa

```
<?PHP
class Budynek
{
    public $nazwa = 'Dowolna wartosc' ;

    public function Wypisz() {
        echo $this->nazwa;
    }
}

$Dom=new Budynek() ;
$Dom->Wypisz() ;
?>
```

PHP5 programowanie obiektowe

Automatyczne ładowanie klas

```
<?PHP
function __autoload($class_name) {
    require_once $class_name . '.php';
}

$Dom=new Budynek();
$Dom->Wypisz();
$Dom->nazwa="Test";
echo "<br>".$Dom->nazwa;
?>
```

Nazwa klasy Budynek w pliku Budynek.php

PHP5 programowanie obiektowe

Referencja

```
<?PHP
class Budynek
{
    public $nazwa = 'Dowolna wartosc';

    public function Wypisz() {
        echo $this->nazwa;
    }
}
$Dom=new Budynek();
$Szopa = $Dom;
$Stajnia =& $Dom;
```

PHP5 programowanie obiektowe

Referencja

```
$Dom->Wypisz();  
$Dom->nazwa="Nazwa domu";  
  
echo "<br> klasa Dom ".$Dom->nazwa;  
echo "<br> klasa Szopa ".$Szopa->nazwa;  
  
$Szopa->nazwa="Nazwa szopy";  
echo "<br> klasa Dom ".$Dom->nazwa;  
echo "<br> klasa Szopa ".$Szopa->nazwa."<br>";  
$Dom=null;  
var_dump($Dom);  
var_dump($Szopa);  
var_dump($Stajnia);  
?>
```

PHP5 programowanie obiektowe

Konstruktor/Destruktor

```
<?PHP
class Budynek
{
    public $nazwa = '';

    function __construct() {
        echo "Konstruktor<br>";
        $this->nazwa = "Brak nazwy";
    }
    function __destruct() {echo "Destruktor<br>";}
    public function Wypisz() {echo $this->nazwa."<br>";}
}
$Dom=new Budynek();
$Dom->Wypisz();
?>
```

PHP5 programowanie obiektowe

Dziedziczenie

```
<?PHP
class Budynek
{
    public $nazwa = 'dowolna';
    function Wypisz(){echo $this->nazwa."<br>";}
}
class Gmach extends Budynek
{
    public $rok='1980';
    function Wypisz(){
        parent::Wypisz();
        echo "<br>".$this->rok."<br>";
    }
}
$Uniwersytet=new Gmach();
$Uniwersytet->Wypisz();
?>
```

PHP5 programowanie obiektowe

Dziedziczenie

```
<?PHP
class Budynek
{
    public $nazwa = 'dowolna';

    function __construct($nazwa) {
        echo "Konstruktor Budynek<br>";
        $this->nazwa = $nazwa;
    }

    function Wypisz() {
        echo "Nazwa: ".$this->nazwa."<br>";
    }
}
```

PHP5 programowanie obiektowe

Zasięg zmiennych

```
<?PHP
class Budynek
{
    public $nazwa = 'dowolna';
    private $koszt = 'koszt';
    protected $wiek = 'wiek';

    function __construct($nazwa) {
        echo "Konstruktor Budynek<br>";
        $this->nazwa = $nazwa;
    }
    function Wypisz() {
        echo "Nazwa: ".$this->nazwa."<br>";
        echo "Koszt: ".$this->koszt."<br>";
        echo "Wiek: ".$this->>wiek."<br>";
    }
}
```

Autor wykładu: A. Dawid, email: dawid@us.edu.pl

PHP5 programowanie obiektowe

Zasięg zmiennych

```
class Gmach extends Budynek
{
    public $rok='1980';
    function __construct($nazwa,$rok) {
        parent::__construct($nazwa);
        echo "Konstruktor Gmach<br>";
        $this->rok = $rok;
    }
    function Wypisz(){
        parent::Wypisz();
        echo "Rok: ".$this->rok."<br>";
        echo "KosztGmach: ".$this->koszt."<br>";
        echo "WiekGmach: ".$this->wiek."<br>";
    }
}
```

PHP5 programowanie obiektowe

Zasięg zmiennych

```
$Uniwersytet=new Gmach("Nasz","1990");
$Uniwersytet->Wypisz();

echo "-----Wywołanie-----<br>";
echo "Nazwa>> ".$Uniwersytet->nazwa."<br>";
echo "Koszt>> ".$Uniwersytet->koszt."<br>";
echo "Wiek>> ".$Uniwersytet->>wiek."<br>";

?>
```

PHP5 programowanie obiektowe

Klasy abstrakcyjne

```
<?PHP
abstract class Budowla
{
    abstract protected function Wartosc();
    public function Wypisz(){
        echo "Wartosc: ".$this->Wartosc()."<br>";
    }
}
class Budynek extends Budowla
{
    protected function Wartosc()
    { return 'Liczba 55';}
}
$Dom=new Budynek();
$Dom->Wypisz();
?>
```

PHP5 programowanie obiektowe

Interfejsy

```
<?PHP

interface Budowanie
{
    public function Ustaw($brygada, $liczba);
}

interface Wyplacanie
{
    public function Kasuj($brygada);
}
```

PHP5 programowanie obiektowe

Interfejsy

```
class Budynek implements Budowanie, Wyplacanie
{
    public $Brygady = array();

    public function Ustaw($brygada, $liczba)
    { $this->Brygady[$brygada]=$liczba; }

    public function Kasuj($brygada)
    { $this->Brygady[$brygada]=null; }

    public function Wypisz($brygada)
    {echo "Brygada: " . $this->Brygady[$brygada] . "<br>"; }
}
```

PHP5 programowanie obiektowe

Interfejsy

```
$Dom=new Budynek();  
$Dom->Ustaw('Tygrysa','20');  
$Dom->Wypisz('Tygrysa');  
$Dom->Kasuj('Tygrysa');  
$Dom->Wypisz('Tygrysa');
```

```
?>
```

PHP5 programowanie obiektowe

Dziedziczenie

```
class Gmach extends Budynek
{
    public $rok='1980';
    function __construct($nazwa,$rok) {
        parent::__construct($nazwa);
        echo "Konstruktor Gmach<br>";
        $this->rok = $rok;
    }
    function Wypisz(){
        parent::Wypisz();
        echo "Rok: ".$this->rok."<br>";
    }
}
$Uniwersytet=new Gmach("Nasz","1990");
$Uniwersytet->Wypisz();
?>
```

Funkcje Klas/Obiektów

`class_exists` - sprawdza czy klasa została zdefiniowana
`get_class_methods` - Zwraca tablicę nazw metod danej klasy
`get_class_vars` - Zwraca tablicę domyślnych właściwości klasy
`get_class` - Zwraca nazwę klasy danego obiektu
`get_declared_classes` - Zwraca tablicę z nazwami zdefiniowanych klas
`get_declared_interfaces` - Zwraca tablicę wszystkich zdefiniowanych klas
`get_object_vars` - Zwraca tablicę asocjacyjną właściwości obiektu
`get_parent_class` - Pobiera nazwę klasy przodka dla obiektu lub klasy

Funkcje Klas/Obiektów

`interface_exists` – Sprawdza czy interfejs został zdefiniowany

`is_a` – Zwraca TRUE jeżeli obiekt jest tej klasy, lub ta klasa jest jednym z jego przodków

`is_subclass_of` – Zwraca TRUE jeżeli klasa jest jednym z przodków obiektu

`method_exists` – Sprawdza czy metoda klasy istnieje

`property_exists` – Sprawdza czy obiekt lub klasa ma właściwości