
PHP



PHP

- PHP to skryptowy, obiektowy język programowania wykorzystywany głównie do tworzenia aplikacji internetowych
- Początki języka sięgają do 1994 r., pierwsza wersja opracowana przez Rasmusa Lerdorfa jako zestaw skryptów służących do monitorowania internautów odwiedzających jego witrynę

PHP

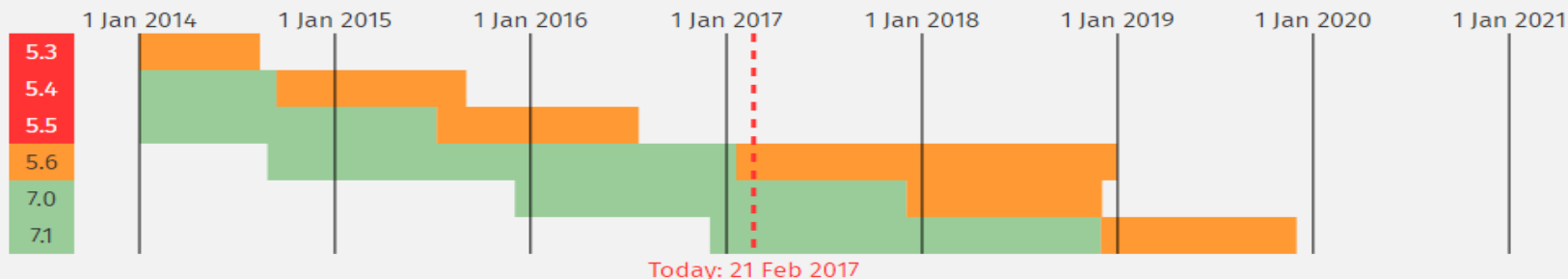
- PHP – pierwotne rozwinięcie skrótu Personal Home Page, aktualnie PHP Hypertext Preprocessor
- PHP jest projektem Open Source – kod źródłowy jest jawnie dostępny i można go swobodnie wykorzystywać w ramach licencji
- Głównym sponsorem Zend Technologies
- Ostatnia stabilna wersja – 7.1.2

PHP

Wsparcie dla poszczególnych wersji (źródło: <http://php.net/supported-versions.php>)

Branch	Initial Release		Active Support Until		Security Support Until	
5.6 *	28 Aug 2014	<i>2 years, 5 months ago</i>	19 Jan 2017	<i>1 month ago</i>	31 Dec 2018	<i>in 1 year, 10 months</i>
7.0	3 Dec 2015	<i>1 year, 2 months ago</i>	3 Dec 2017	<i>in 9 months</i>	3 Dec 2018	<i>in 1 year, 9 months</i>
7.1	1 Dec 2016	<i>2 months ago</i>	1 Dec 2018	<i>in 1 year, 9 months</i>	1 Dec 2019	<i>in 2 years, 9 months</i>

Or, visualised as a calendar:



PHP

- Zalety PHP:
 - wydajność,
 - duży zestaw wbudowanych bibliotek pozwalających tworzyć zaawansowane funkcjonalności za pomocą kilku linii kodu,
 - zerowy koszt – rozwiązanie Open Source,
 - (względnie) łatwość nauki,
 - przenośność – działa na wszystkich systemach operacyjnych,
 - wiele frameworków (Zend, Symfony, CakePHP, CodeIgniter, Yii2 i wiele innych) ułatwiających kodowanie

PHP

- Wady:
 - przestarzałe (bo innych argumentów konkurencji brak)

PHP

- Główni konkurenci: Perl, Microsoft ASP.NET, Ruby, Java Server Pages, Cold Fusion
- Do poprawnego działania skryptów PHP potrzebujemy - samego PHP, serwera WWW (Apache lub IIS) i przeglądarki WWW
- Kod PHP jest interpretowany przez serwer WWW i przekazywany do przeglądarki
- Dokumentacja: <http://www.php.net/>

Praca z serwerem

- Najpopularniejsze serwery WWW to:
 - Apache – rozwiązanie Open Source
 - IIS – Microsoft Internet Information Services
- Możemy skorzystać z serwera zdalnego (usługa hostingu, zazwyczaj płatna, ale są też rozwiązania bezpłatne) lub lokalnego (rozwiązanie bezpłatne)

Praca z serwerem

- W przypadku Apache mamy dostępne aplikacje dla Windows, które kilkoma kliknięciami udostępniają nam całą platformę WWW (serwer WWW, PHP i serwer baz danych MySQL/MariaDB), np. XAMPP, WAMP, ale należy je wykorzystywać tylko do pracy nad projektem
- W środowisku produkcyjnym najpopularniejszym rozwiązaniem jest LAMP – Linux, Apache, MySQL, PHP
- IIS jest również rozwiązaniem bezpłatnym, przy czym każdy element platformy instalujemy oddzielnie, można wykorzystywać w środowisku produkcyjnym

Praca z serwerem

- Instalacja XAMPP
 - pobieramy plik instalacyjny ze strony <https://www.apachefriends.org/pl/index.html> lub wersję portable <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.1.1/>
 - instalujemy plik lub rozpakowujemy archiwum do katalogu x:\xampp (gdzie x to litera dowolnego dysku)
 - uruchamiamy x:\xampp\xampp-control.exe z uprawnieniami administratora

Praca z serwerem

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

Config

11:29:52 [main] Control Panel Version: 3.2.2 [Compiled: Nov 12th 2015]
11:29:52 [main] Running with Administrator rights - good!
11:29:52 [main] XAMPP Installation Directory: "d:\xampp\
11:29:52 [main] Checking for prerequisites
11:29:52 [main] All prerequisites found
11:29:52 [main] Initializing Modules
11:29:52 [main] The FileZilla module is disabled
11:29:52 [main] The Mercury module is disabled
11:29:52 [main] Starting Check-Timer
11:29:52 [main] Control Panel Ready

Praca z serwerem

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	996 2968	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	5488	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config
 Netstat
 Shell
 Explorer
 Services
 Help
 Quit

11:29:52 [main] All prerequisites found
11:29:52 [main] Initializing Modules
11:29:52 [main] The FileZilla module is disabled
11:29:52 [main] The Mercury module is disabled
11:29:52 [main] Starting Check-Timer
11:29:52 [main] Control Panel Ready
11:30:52 [Apache] Attempting to start Apache app...
11:30:52 [Apache] Status change detected: running
11:30:55 [mysql] Attempting to start MySQL app...
11:30:55 [mysql] Status change detected: running

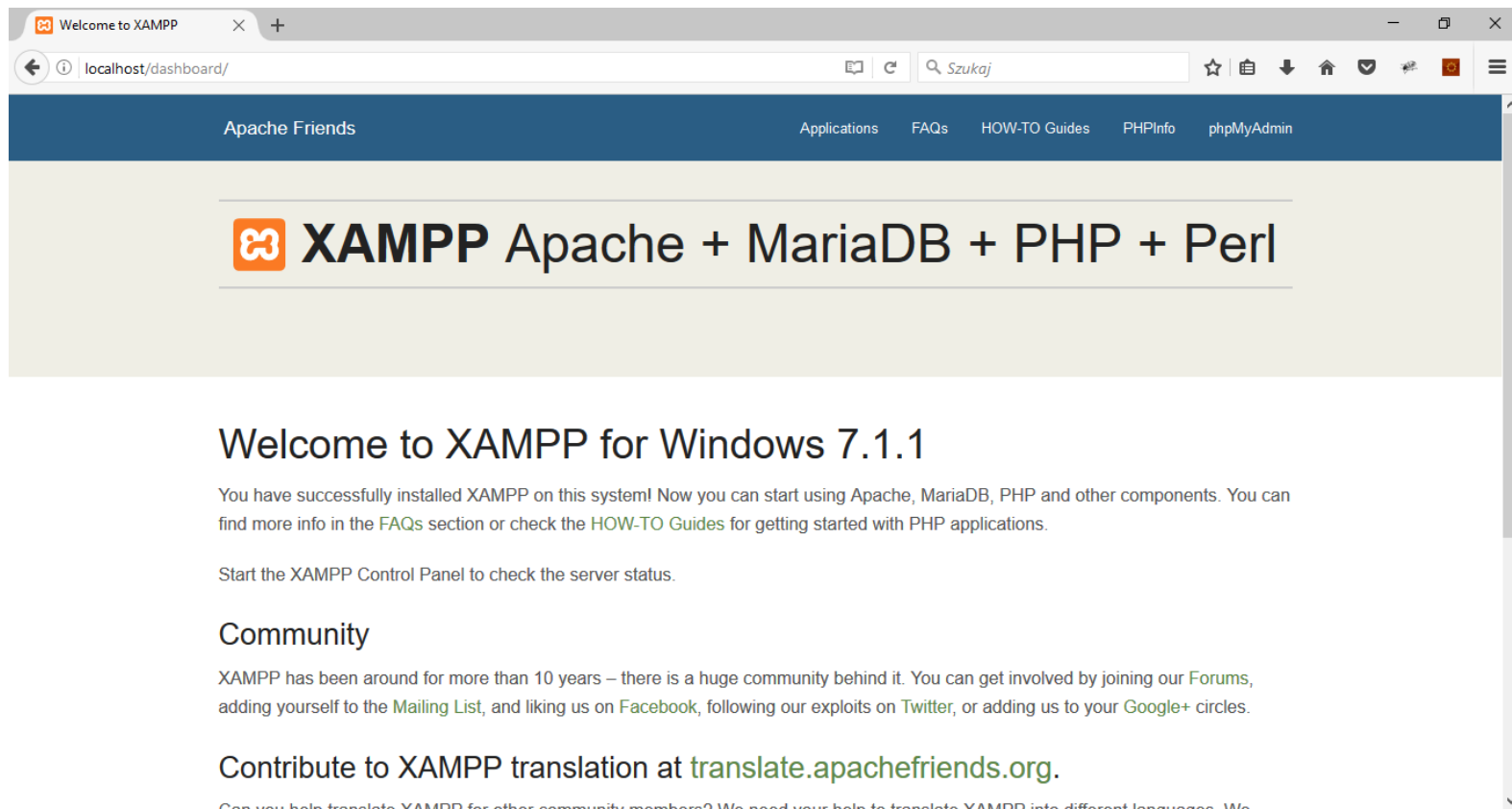
Praca z serwerem

- Jeśli pojawią się problemy z uruchomieniem serwerów:
 - uruchamiamy plik `x:\xampp\setup_xampp.bat`
 - sprawdzamy czy port 80 nie jest aktualnie używany przez inną aplikację (np. Skype) i albo wyłączamy tą aplikację, albo zmieniamy domyślny port serwera WWW na, np. 8080 w Actions -> Config -> Apache (`httpd.conf`) -> linia z komendą `Listen 80`

Praca z serwerem

- Aby uruchomić stronę startową serwera należy kliknąć `Admin` na poziomie `Apache` lub wpisać w przeglądarce adres `localhost/127.0.0.1`
- Wszystkie pliki, które mają zostać przetworzone przez serwer WWW, umieszczamy w folderze `x:\xampp\htdocs`

Praca z serwerem



The screenshot shows a web browser window with the address bar displaying 'localhost/dashboard/'. The page has a dark blue header with the text 'Welcome to XAMPP' and navigation links: 'Applications', 'FAQs', 'HOW-TO Guides', 'PHPInfo', and 'phpMyAdmin'. Below the header, the main content area features the XAMPP logo (an orange square with a white 'X') followed by the text 'XAMPP Apache + MariaDB + PHP + Perl'. The page is titled 'Welcome to XAMPP for Windows 7.1.1' and contains the following text:

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), adding yourself to the [Mailing List](#), and liking us on [Facebook](#), following our exploits on [Twitter](#), or adding us to your [Google+](#) circles.


Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We

Praca z serwerem

Configure Virtual Hosts x phpinfo()

localhost/dashboard/phpinfo.php Szukaj

PHP Version 7.1.1 

System	Windows NT SAMSUNG 10.0 build 14393 (Windows 10) i586
Build Date	Jan 18 2017 18:37:40
Compiler	MSVC14 (Visual C++ 2015)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=.obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20160303
PHP Extension	20160303
Zend Extension	320160303
Zend Extension Build	API320160303,TS,VC14
PHP Extension Build	API20160303,TS,VC14
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring

Praca z serwerem

- Ustawienia PHP zawarte są w pliku konfiguracyjnym `x:\xampp\php\php.ini`
- Przy podstawowych zadaniach informacje zawarte w `php.ini` nie są potrzebne
- Przy serwerach zdalnych użytkownik zazwyczaj nie ma możliwości zmiany ustawień PHP (można próbować dodać w katalogu głównym aplikacji swój plik `php.ini` lub `.htaccess`)

Podstawy PHP

- Aby kod PHP mógł zostać poprawnie zinterpretowany przez serwer WWW, plik musi posiadać rozszerzenie .php
- Kod PHP musi znaleźć się pomiędzy znacznikiem otwierającym i zamykającym
- Należy bezwzględnie przestrzegać składni PHP, w przeciwnym razie po napotkaniu pierwszego błędu kod przestanie być wykonywany

Podstawy PHP

- 4 style znaczników PHP:
 - styl XML: `<?php ... ?>` - zalecany
`<?php echo 'Hello World!'; ?>`
 - styl krótki: `<? ... ?>` - niezalecany, wymaga zmiany ustawień PHP (`short_open_tag`), może zostać wyłączony przez administratora serwera
`<? echo 'Hello World!'; ?>`

Podstawy PHP

- styl scripts: `<script language='php'> ...
</script>`
`<script language='php'> echo 'Hello
World!'; </script>` – **UWAGA: usunięty w
PHP 7.0.0**
- styl ASP: `<% ... %>` - wymaga zmiany ustawień
`asp_tags` – **UWAGA: usunięty w PHP 7.0.0**
`<% echo 'Hello World!'; %>`

Podstawy PHP - zadanie

Zadanie 1 PHP.pdf

Podstawy PHP - komentarze

- PHP udostępnia 3 sposoby na tworzenie komentarzy:
 - komentarz wielowierszowy pomiędzy znacznikami
`/**/
/* Komentarz */`
 - komentarz jednowierszowy za znacznikiem `//`
`// Komentarz`
 - komentarz jednowierszowy za znacznikiem `#`
`# Komentarz`

Podstawy PHP - zmienne

- Zmienna to kontener, który może przechowywać jakąś wartość. Każda zmienna ma swoją nazwę (identyfikator) oraz miejsce przechowywania w pamięci komputera
- W PHP zmienne deklarujemy w następujący sposób:
`$identyfikator`
- Nazwa zmiennej NIE może rozpoczynać się od liczby i innego znaku specjalnego oprócz `_`
- PHP zwraca uwagę na wielkość znaków (`$var` jest inną zmienną niż `$Var`)

Podstawy PHP - zmienne

`$zmienna` – dobrze

`$_var` - dobrze

`$5_zmienna` – źle

`$.zmienna` – źle

`$jakasZmienna` – dobrze

`$jakas_zmienna` - dobrze

Podstawy PHP - zmienne

- PHP charakteryzuje się słabym typowaniem zmiennych (dynamicznym), co oznacza, że w trakcie wykonywania programu zmienna może przechowywać różne typy wartości (np. na początku może przechowywać liczbę, a za chwilę tekst)
- Większość języków programowania wymaga jasnego zdefiniowania typu zmiennej

Podstawy PHP - zmienne

- Typy zmiennych:
 - `integer` – liczba całkowita, np. `5`
 - `float` – liczba zmiennoprzecinkowa, np. `5.68`
 - `string` – łańcuch znaków, np. `"Hello World!"`
 - `array` – tablica do przechowywania wielu danych
 - `boolean` – zmienna logiczna, przyjmująca wartość `true (1)` lub `false (0)`

Podstawy PHP - zmienne

- Podstawienie wartości do zmiennej realizujemy za pomocą znaku =, np.:

```
$car = "Opel";
```

```
$car2 = 'Mercedes';
```

```
$number = 5;
```

```
$car = $number;
```

```
$array = array(7, 9, 10);
```

```
$array2 = [7, 9, 10];
```

Podstawy PHP – ciągi znaków

- Ciągi znaków możemy umieszczać pomiędzy apostrofami ('ciąg znaków') lub w cudzysłowie ("ciąg znaków")
- Ciąg znaków umieszczony pomiędzy apostrofami traktowany jest jak surowe dane
- Jeśli użyjemy cudzysłowów, PHP jest w stanie wyszukać zmienną i podstawić jej wartość

Podstawy PHP – ciągi znaków

```
$car = "Opel";  
echo 'Kupiłem samochód marki $car';  
//=>Kupiłem samochód marki $car  
  
echo "Kupiłem samochód marki $car";  
//=>Kupiłem samochód marki Opel
```

Podstawy PHP - operatory

- Operatory pozwalają wykonywać operacje na zmiennych
- Wyróżniamy następujące typy operatorów:
 - arytmetyczne:
 - + suma, np. `$a + $b, $c = 2 + 5;`
 - - różnica, np. `$a - $b, $c = 2 - 5;`
 - * iloczyn, np. `$a * $b, $c = 2 * 5;`
 - / iloraz, np. `$a / $b, $c = 2 / 5;`
 - % modulo (reszta z dzielenia), np. `$a % $b, $c = 5%2;`

Uwaga: jeśli nie uda się interpreterowi przekonwertować zmiennej na liczbę, zostanie użyte 0.

Podstawy PHP - operatory

- konkatencji (łączenia łańcuch znaków): . (kropka)

```
$a = "Zakład";
```

```
$b = "Fryzjerski";
```

```
echo $a . ' ' . $b; // => Zakład  
Fryzjerski
```

- przypisania: =

```
$a = 17;
```

```
$b = $a;
```

```
echo $b; // => 17
```

Podstawy PHP - operatory

- łączonego przypisania: `+=`, `-=`, `*=`, `/=`, np.:

```
$a = 3;  
$a += $a;  
echo $a; // => 6
```

- post- preinkrementacji, post- predekrementacji: `++`, `--`, np.:

```
$a = 7;  
echo $a++; // => 7;  
echo $a; // => 8  
$b = 3;  
echo --$b; // => 2  
echo $b; // => 2
```


Podstawy PHP - operatory

- porównań: == (równy), != (różny), > (większy), < (mniejszy), >= (większy, bądź równy), <= (mniejszy, bądź równy), === (identyczny), np.:

```
$a = 3;
```

```
$b = 7;
```

```
echo $a == $b; // =>
```

```
echo $a <= $b; // => 1
```

```
$a = 5;
```

```
$b = '5';
```

```
echo $a == $b; // => 1 (dlaczego?)
```

```
echo $a === $b; // => (dlaczego?)
```

Podstawy PHP - operatory

- logiczne: `&&` (koniunkcja), `||` (alternatywa), `!` (zaprzeczenie), np.:

```
$a = 10;
```

```
$start = 0;
```

```
$end = 20;
```

```
echo $a >= $start && $a <= $end; // => ??
```

```
$b = 10;
```

```
echo $b == $a || $b == $end; // => ??
```

Podstawy PHP - operatory

- tłumienia błędów: @, np.:

```
$a = 7;
```

```
$b = 0;
```

```
@ $c = $a / $b;
```

```
echo $c; // => ??
```

Podstawy PHP – obsługa formularzy

- Za pomocą PHP można uzyskać dostęp do każdego pola formularza
- W zależności od tego, czy formularze przekazywane są metodą POST czy GET, dostęp do pól formularza realizowany jest następująco:

Podstawy PHP – obsługa formularzy

- **POST:**

```
$_POST['name'] lub  
HTTP_POST_VARS['name']
```

- **GET:**

```
$_GET['name'] lub  
HTTP_GET_VARS['name']
```

Podstawy PHP – obsługa formularzy

```
<form method="post" action="post.php">  
    <input type="text" name="surname">  
</form>
```

```
$surname = $_POST['surname'];
```

Adres strony zmieni się na
<http://localhost/post.php>

Podstawy PHP – obsługa formularzy

```
<form method="get" action="get.php">  
    <input type="text" name="surname">  
</form>
```

```
$surname = $_GET['surname'];
```

Adres strony zmieni się na

<http://localhost/get.php?surname=kowalski>

Podstawy PHP - zadanie

Zadanie 2 PHP.pdf

Podstawy PHP – instrukcje warunkowe

- Instrukcje warunkowe dają możliwość wykonania kodu w zależności od spełnienia (bądź nie) warunku logicznego
- Dzięki temu nasz program może dynamicznie "podejmować decyzje" według ściśle określonych reguł

Podstawy PHP – instrukcje warunkowe

- Wyróżniamy następujące instrukcje warunkowe:
 - `if` – jeżeli – sprawdza czy warunek to prawda czy fałsz,

```
if (warunek) {  
    kod_do_wykonania;  
}
```

Podstawy PHP – instrukcje warunkowe

```
$a = 7;
```

```
if ($a > 5) {  
    echo "$a jest większe od 5";  
} // => 7 jest większe od 5
```

Podstawy PHP – instrukcje warunkowe

- `else` – w przeciwnym razie (rozszerzenie `if`)

```
if (warunek) {  
    kod_do_wykonania;  
} else {  
    kod_do_wykonania;  
}
```

Podstawy PHP – instrukcje warunkowe

```
$a = 3;  
if ($a > 5) {  
    echo "$a jest większe od 5";  
} else {  
    echo '$a nie jest większe od 5';  
} // => $a jest mniejsze od 5  
skrótowa wersja:  
warunek ? kod dla true : kod dla false;
```

Podstawy PHP – instrukcje warunkowe

- `elseif` – w przeciwnym razie, jeżeli (rozszerzenie `if`)

```
if (warunek) {  
    kod_do_wykonania;  
} elseif (warunek) {  
    kod_do_wykonania;  
}
```

Podstawy PHP – instrukcje warunkowe

```
$a = 3;  
if ($a > 5) {  
    echo "$a jest większe od 5";  
} elseif ($a == 3) {  
    echo '$a = 3';  
} else {  
    echo 'Pogubiłem się...';  
} //=> $a = 3
```

Podstawy PHP – instrukcje warunkowe

- `switch` – w odróżnieniu od `if` sprawdza czy warunek przyjmuje konkretną wartość

```
switch (warunek) {  
    case wartosc_1:  
        kod do wykonania;  
        break;  
  
    ...  
  
    default:  
        kod do wykonania;  
        break;
```


Podstawy PHP – instrukcje warunkowe

```
$a = 'samochód';  
switch ($a){  
    case 1:  
        echo '$a = 1';  
        break;  
    case 'samochód':  
        echo 'car';  
        break;  
    default:  
        echo 'nie wiem nic';  
        break; }  

```

Podstawy PHP – zadanie

Zadanie 3 PHP.pdf

Podstawy PHP - pętle

- W sytuacjach, gdy chcemy wykonać ten sam kod dla wielu elementów, bądź też wiele razy na tym samym elemencie, warto skorzystać z pętli
- Dzięki temu zmniejszamy ilość kodu i oszczędzamy czas

Podstawy PHP - pętle

- Wyróżniamy 3 podstawowe rodzaje pętli:
 - `while` – kod jest wykonywany, dopóki warunek sprawdzający jest spełniony, np.:

```
$i = 10;
while ($i >=0) {
    echo "<p>$i</p>";
    $i--;
}
```

Podstawy PHP - pętle

- `do ... while` – podobnie jak `while`, przy czym pętla wykona się przy najmniej 1 raz, ponieważ warunek sprawdzany jest na końcu, np.

```
$i = 0;
do {
    echo "<p>$i</p>";
    $i++; }
while ($i <10);
```

Podstawy PHP - pętle

- `for (licznik; warunek; działanie_na_liczniku)` – pętla, która wykona się z góry określoną ilość razy, np.

```
$i = 0;
for ($i; $i <=10; $i++) {
    echo "<p>$i</p>";
}
```

Podstawy PHP - tablice

- Tablica jest strukturą, która w przeciwieństwie do zmiennej, pozwala przechowywać zbiory danych
- Tablica może być jednowymiarowa lub wielowymiarowa. W praktyce najczęściej wykorzystuje się tablice dwu- trzywymiarowe
- Każdy element tablicy może być wartością lub inną tablicą

Podstawy PHP - tablice

- Wyróżniamy 2 rodzaje tablic:
 - indeksowane numerycznie – indeksem/kluczem są kolejne (ale niekoniecznie) liczby, zaczynając od 0
 - asocjacyjne – indeksowane dowolnym kluczem, najczęściej słowem

Podstawy PHP - tablice

- Tablicę możemy zadeklarować na 2 sposoby:
 - `$nazwa = array(el_0, el_1, ..., el_n), np.:`
`$cars = array("Vectra", "Sprinter",
"105");`
`//jednowymiarowa tablica indeksowana numerycznie`
`$cars2 = array(
 array("Vectra", "Sprinter", "105"),
 array("Scania", "DAF", "Iveco")
);`

Podstawy PHP - tablice

- `$nazwa = [el_0, el_1, ..., el_n], np.:`
`$cars = ["Vectra", "Sprinter", "105"];`
`//jednowymiarowa tablica indeksowana`
`numerycznie`
- **Dostęp do elementu tablicy realizowany poprzez odwołanie do jego indeksu:**
`echo $cars[0]; => Vectra`
`echo $cars[2]; => 105`

Podstawy PHP - tablice

```
$cars2 = [  
    ["Vectra", "Sprinter", "105"],  
    ["Scania", "DAF", "Iveco"]  
]; //tablica dwuwymiarowa  
echo $cars2[1][1]; //=> DAF  
echo $cars2[2][2]; //=> ???
```

Podstawy PHP - tablice

```
$assoc_cars = ["Opel" =>  
"Vectra", "Mercedes" =>  
"Sprinter", "Syrena" => "105"];  
//jednowymiarowa tablica  
asocjacyjna  
echo $assoc_cars["Syrena"]; // =>  
105;
```

Podstawy PHP - tablice

```
$assoc_cars2 = [  
    "osobowe" => [  
        "Opel" => "Vectra",  
        "Mercedes" => "Sprinter",  
        "Syrena" => "105"  
    ],  
    "dostawcze" => [  
        "DAF", "MAN", "Scania"  
    ]  
]  
  
echo $assoc_cars2["osobowe"]["Opel"]; // => Vectra  
echo $assoc_cars2["dostawcze"][1]; // => MAN
```

Podstawy PHP - tablice

- Zmiana wartości elementu tablicy lub dodanie elementu do tablicy realizowane są poprzez operator przypisania:

```
$num = [1, 2, 3];
```

```
echo $num[2]; // => 3
```

```
$num[2] = 5;
```

```
echo $num[2]; // => 5
```

Podstawy PHP - tablice

```
$num = [1, 2, 3];
```

```
$num[4] = 7;
```

```
echo $num[4]; // => 7
```

```
echo $num[3]; // => ???
```

```
$num[] = 15;
```

```
//pod jakim indeksem znajdzie się nowa  
wartość?
```

Podstawy PHP - tablice

- Aby wyświetlić wszystkie elementy tablicy, należy skorzystać z pętli
- Do tego celu najlepiej wykorzystać pętlę `for` lub `foreach` (specjalna pętla dla tablic)

Podstawy PHP - tablice

```
$num = [1, 2, 3];  
for ($i=0; $i<3; $i++) {  
    echo '<p>'.$num[$i]. '</p>';  
} // =>  
//1  
//2  
//3
```

Podstawy PHP - tablice

- Jeśli ilość elementów tablicy nie jest nam znana, lub może się dynamicznie zmieniać w trakcie wykonywania kodu, warto skorzystać z funkcji `count()`, która zlicza elementy tablicy:

```
$num = [1, 2, 3];  
echo count($num); // => 3
```

Podstawy PHP - tablice

```
$array = [????];  
for ($i=0; $i < count($array); $i++) {  
    echo $array[$i];  
}
```

UWAGA: w tym przypadku `count()` zwróci tylko liczbę elementów pierwszego poziomu (jeśli `$array` będzie zawierała 1 element, który będzie tablicą o `n` elementach, to zwróci wartość 1) –

<http://php.net/manual/pl/function.count.php>

Podstawy PHP - tablice

- Użycie pętli `foreach` jest znacznym ułatwieniem przy operacjach na tablicach:

```
foreach ($array as $element) {  
    echo $element; } lub  
foreach ($array as $key =>  
$value) {  
    echo $key . '=>' . $value; }
```

Podstawy PHP - tablice

```
$arr = [5, "samochodów",  
"chciałbym", "mieć"];  
foreach ($arr as $el) {  
    echo $el. ' ';  
} => // 5 samochodów chciałbym  
mieć
```

Podstawy PHP - tablice

```
$num = [5, 6, 7, 8];  
foreach ($num as &$n) { //& - referencja  
    $n *=2;  
    unset($n); //czyści zmienną $n  
}  
foreach ($num as $n) {  
    echo $n.', ';  
} // => 10, 12, 14, 16,
```

Podstawy PHP - tablice

- Przydatne funkcje przy tablicach (i nie tylko):
 - `is_array($zmienna)` – sprawdza, czy zmienna jest tablicą, zwraca wartość `true` lub `false`
 - `range(start, end, jump)` – inicjuje tablicę liczbami zaczynając od wartości `start` do `end`. `jump` jest opcjonalny, określa przeskok między wartościami
 - `is_set($zmienna)` – sprawdza, czy zmienna została zainicjowana jakąś wartością, zwraca `true` lub `false`
 - `print_r($zmienna)/var_dump($zmienna)` – wyświetlają informacje o zmiennej (typ, wartość),
 - `is_numeric($zmienna)` – sprawdza, czy zmienna jest typu liczbowego

Podstawy PHP - tablice

Zadanie 4 PHP.pdf

Podstawy PHP – operacje na plikach

- PHP umożliwia zapisanie przesłanych przez użytkownika strony danych
- Jednym ze sposobów jest zapis do pliku, przy czym trzeba mieć na uwadze, że jest to rozwiązanie obciążone dużymi ograniczeniami w stosunku do baz danych

Podstawy PHP – operacje na plikach

- Czasami trzeba jednak obsłużyć pliki przesłane na serwer przez użytkownika (np. plik CSV z danymi kontaktowymi albo cennik produktów)
- Opisane w dalszej części funkcje mają zastosowanie w obydwu przypadkach

Podstawy PHP – operacje na plikach

- Etapy zapisywania danych do pliku:
 1. Otwarcie pliku lub jego utworzenie jeśli nie istnieje
 2. Zapis danych pliku
 3. Zamknięcie pliku

Podstawy PHP – operacje na plikach

- Etapy odczytywania danych z pliku:
 1. Otwarcie pliku, a jeśli nie istnieje, wyświetlenie komunikatu
 2. Odczyt danych z pliku
 3. Zamknięcie pliku

Podstawy PHP – operacje na plikach

- W zależności od potrzeb, plik może zostać otworzony w jednym z następujących trybów:
 - `r` – odczyt pliku od jego początku
 - `r+` - odczyt i zapis do pliku od jego początku
 - `w` – zapis do pliku od jego początku (UWAGA: jeśli plik istnieje, bieżąca zawartość zostanie nadpisana. Jeśli nie istnieje – zostanie utworzony)

Podstawy PHP – operacje na plikach

- `w+` - odczyt i zapis od początku pliku (UWAGA: jak poprzednio)
- `x` - zapis do pliku od jego początku, jeśli nie istnieje to zostanie wyświetlone ostrzeżenie
- `x+` - odczyt i zapis od początku pliku, jeśli nie istnieje to zostanie wyświetlone ostrzeżenie

Podstawy PHP – operacje na plikach

- `a` – dodawanie zawartości na końcu pliku, jeśli nie istnieje – zostanie utworzony
- `a+` - odczyt i dodawanie zawartości na końcu pliku, jeśli nie istnieje – zostanie utworzony
- `b` – tryb binarny, zapewniający zgodność z używanym systemem operacyjnym, stosowany z jednym z powyższych trybów, zalecany
- `t` – tryb tekstowy, dostępny jedynie w Windows, stosowany z jednym z powyższych trybów, zalecany

Podstawy PHP – operacje na plikach

- Otworzenie pliku umożliwia funkcja `fopen()`, która w podstawowej wersji przyjmuje jako argumenty ścieżkę do pliku i tryb otwarcia

```
fopen ('ścieżka/nazwa.txt', 'tryb')
```

```
fopen ('ścieżka', 'tryb',  
'include_path=true')
```

```
fopen ('ścieżka', 'tryb',  
'include_path=true', 'kontekst')
```


Podstawy PHP – operacje na plikach

- Jeśli otworzenie pliku powiedzie się, funkcja `fopen()` zwróci uchwyt/wskaźnik do pliku
- W przeciwnym razie zwróci wartość `false` i zostanie wyświetlone ostrzeżenie

```
$file_handler = fopen('../plik.txt', 'ab');  
@$file_handler = fopen('z/plik.txt', 'ab');
```

Podstawy PHP – operacje na plikach

- Bardzo ważne jest dokładne określenie ścieżki pliku
- Jeśli katalog, w którym chcemy otworzyć plik, nie istnieje lub katalog ma wyłączone prawa do zapisu, to zawsze otrzymamy błąd
- Ponadto należy pamiętać, że błędne określenie ścieżki do pliku, może mieć wpływ na stabilność i bezpieczeństwo serwera
- Po zakończeniu pracy z plikiem należy go zamknąć za pomocą funkcji `fclose (uchwyt_do_pliku);`

Podstawy PHP – operacje na plikach

- Przy określaniu ścieżki pliku warto skorzystać ze zmiennej środowiskowej
`$_SERVER['DOCUMENT_ROOT']`
- Zwraca ona ścieżkę do katalogu publicznego
- Dzięki temu łatwo określimy ścieżkę do pliku względem katalogu publicznego a nie pliku, w którym staramy się uzyskać dostęp

Podstawy PHP – operacje na plikach

```
echo $_SERVER['DOCUMENT_ROOT']; // =>  
X:/xampp/htdocs  
  
$public = $_SERVER['DOCUMENT_ROOT'];  
fopen("$public/../../zamowienia.txt",  
"ab");
```

Wrażliwe dane zawsze należy umieszczać poza katalogiem publicznym. W przeciwnym razie, jeśli komuś uda się odgadnąć ścieżkę do pliku – uzyska do niego dostęp.

Podstawy PHP – operacje na plikach

- Aby zapisać dane do pliku, należy użyć funkcji `fwrite()` (lub aliasu `fputs()`)
- `fwrite()` w najprostszej postaci przyjmuje jako argumenty uchwyt do pliku (zwracany przez `fopen()`) oraz ciąg danych do zapisu

```
fwrite($file_handler, $data)
```

```
fwrite($fh, $data, strlen($data))
```

Podstawy PHP – operacje na plikach

```
$public = $_SERVER['DOCUMENT_ROOT'];  
$handler = fopen("$public/info.php", "ab");  
$content = "<?php phpinfo() ?>";  
$write = fwrite($handler, $content);  
if ($write) {  
    echo "Zapis do pliku powiódł się";  
} else {  
    echo "Nie udało się zapisać danych do pliku";  
}  
fclose($handler);
```

Podstawy PHP – operacje na plikach

- Alternatywnie możemy skorzystać z funkcji `file_put_contents()`, która wykonuje po kolei operacje otwarcia pliku w trybie `'wb'` (!), zapisu danych i zamknięcia pliku

```
file_put_contents(ścieżka, dane)
```

```
file_put_contents(ścieżka, dane,  
flaga, kontekst)
```

Podstawy PHP – operacje na plikach

- Do odczytu danych z pliku możemy wykorzystać następujące funkcje:
 - `fgets(uchwyt_do_pliku)` – pobiera pojedynczą linię z pliku
 - `fgetss(uchwyt_do_pliku)` – pobiera pojedynczą linię z pliku i oczyszcza ją z tagów HTML i PHP

Podstawy PHP – operacje na plikach

- `fgetc (uchwyt)` – pobiera pojedyncze znaki
- `fgetcsv (uchwyt)` – przekształca wiersze w pola pliku CSV (zwracana jest tablica)
- `file_get_contents (ścieżka_do_pliku)`
 - zwraca całą zawartość pliku jako string
- `feof (uchwyt)` – zwraca `false`, gdy napotka koniec pliku

```
$public = $_SERVER['DOCUMENT_ROOT'];  
$fh = fopen("$public/../../dane.txt", 'r');  
while (!feof($fh)) {  
    $wiersze[] = fgets($fh);  
}  
foreach ($wiersze as $wiersz) {  
    echo $wiersz.'<br>';  
}
```

Podstawy PHP – operacje na plikach

- W przypadku współdzielenia pliku (możliwość jednoczesnego dostępu do pliku przez wielu użytkowników) może dojść do konfliktu związanego z dostępem do niego
- Zachowanie się skryptu będzie wtedy nieprzewidywalne, dane mogą być zapisywane w losowej kolejności

Podstawy PHP – operacje na plikach

- Rozwiązaniem tego problemu jest funkcja `flock()`, która blokuje dostęp do pliku na czas pracy z nim

`flock(uchwyt, operacja)`

- `flock()` nie rozwiązuje jednak problemu pierwszeństwa dostępu do pliku

Podstawy PHP – operacje na plikach

- `flock()` daje możliwość wykonania 4 wariantów operacji na pliku:
 - `LOCK_SH` – blokada zapisu, możliwy odczyt
 - `LOCK_EX` – blokada zapisu i odczyt – dostęp wyłączny
 - `LOCK_UN` – odblokowanie pliku
 - `LOCK_NB` – do użycia z wcześniejszymi opcjami, plik jest dostępny dopóki nie zostanie skutecznie zablokowany (domyślnie już podjęcie próby zablokowania blokuje plik)

```
$public = $_SERVER['DOCUMENT_ROOT'];  
$handler = fopen("$public/../../dane.txt", "ab");  
$lock = flock($handler, LOCK_EX | LOCK_NB);  
$content = "Bardzo ważne dane";  
if ($handler && $lock) {  
    $write = fwrite($handler, $content);  
    if ($write) {  
        echo "Zapis do pliku powiódł się";  
    } else {  
        echo "Nie udało się zapisać danych do pliku";  
    }  
    flock($handler, LOCK_UN); fclose($handler);  
} else {  
    echo "Nie można uzyskać dostępu do pliku";  
}
```

```
$public = $_SERVER['DOCUMENT_ROOT'];  
$content = "Bardzo ważne dane";  
$write = file_put_contents("$public/../../dane.txt",  
FILE_APPEND | LOCK_EX);  
if ($write) {  
    echo "Zapis do pliku powiódł się";  
} else {  
    echo "Nie udało się zapisać danych do pliku";  
}
```

Podstawy PHP – operacje na plikach

Zadanie 5 PHP.pdf

Podstawy PHP – manipulowanie ciągami

- Dzięki wbudowanej w PHP funkcjom możemy w łatwy sposób przetwarzać długie ciągi tekstu w celu ich odpowiedniego formatowania, oczyszczania ze złośliwego kodu lub wyszukiwania interesujących nas danych

Podstawy PHP – manipulowanie ciągami

- Wybrane funkcje formatujące ciągi:
 - `trim(ciąg)` – usuwa białe znaki (spacje, tabulatory, koniec bieżącej linii) po lewej i prawej stronie ciągu
 - `ltrim(ciąg)` i `rtrim(ciąg)` – modyfikacja `trim()`, przy czym usuwa białe znaki odpowiednio po lewej lub prawej stronie ciągu

Podstawy PHP – manipulowanie ciągami

- `nl2br(ciąg)` – zamienia znak końca linii na znacznik `
`, który są w stanie zinterpretować przeglądarki
- `strtoupper(ciąg)` – zamienia wszystkie litery na duże (przydatne przy wyszukiwaniu konkretnych ciągów znaków)
- `strtolower(ciąg)` – zamienia wszystkie litery na małe
- `ucfirst(ciąg)` – pierwsza litera duża
- `ucwords(ciąg)` – duża pierwsza litera każdego wyrazu

Podstawy PHP – manipulowanie ciągami

- `addslashes (ciąg)` – dodaje symbol ucieczki `\` przed wszystkim znakami ustalonymi jako kontrolne (czyli znakami `'`, `"`, `NUL`) – bardzo ważne przy składowaniu w bazie danych wprowadzonych przez użytkownika danych
- `stripslashes (ciąg)` – odwraca działanie `addslashes ()`

Podstawy PHP – manipulowanie ciągami

- `strcmp(ciąg1, ciąg2)` – porównanie dwóch ciągów z włączoną czułością na wielkość liter
- `strcasecmp(ciąg1, ciąg2)` – porównanie dwóch ciągów bez zwracania uwagi na wielkość liter
- `strnatcmp(ciąg1, ciąg2)` – porównanie dwóch ciągów i wybranie mniejszego pod względem leksykograficznym, ze zwróceniem uwagi na wielkość liter (algorytm naturalny)

Podstawy PHP – manipulowanie ciągami

- `strnatcasecmp(ciąg1, ciąg2)` – porównanie dwóch ciągów i wybranie mniejszego pod względem leksykograficznym, bez zwracania uwagi na wielkość liter
- `strlen(ciąg)` – długość ciąg
- `explode(delimiter, ciąg)` – rozbija w miejscu wystąpienia delimitera ciąg znaków na elementy i umieszcza je w tablicy

Podstawy PHP – manipulowanie ciągami

```
$csv = 'Marcin;Pikul;adres@email.com;600700800';  
$e_csv = explode(';', $csv);  
print_r($e_csv);  
//=>Array ( [0] => Marcin [1] => Pikul [2] =>  
adres@email.com [3] => 600700800 )
```

Podstawy PHP – manipulowanie ciągami

- `implode(łącznik, tablica)` –
odwrotność do `explode`
- `strstr(ciąg, punkt_kontrolny)` –
wyszukuje pierwszy punkt kontrolny w ciągu i
zwraca wszystko jego prawej strony łącznie z nim
- `stristr(ciąg, punkt_kontrolny)` –
identycznie jak `strstr()`, przy czym nie zwraca
uwagi na wielkość znaków

Podstawy PHP – manipulowanie ciągami

- `strchr(ciąg, punkt_kontrolny)` –
wyszukuje ostatni punkt kontrolny w ciągu i
zwraca wszystko jego prawej strony łącznie z nim
- `strichr(ciąg, punkt_kontrolny)` –
identycznie jak `strchr()`, przy czym nie zwraca
uwagi na wielkość znaków

Podstawy PHP – manipulowanie ciągami

- `strpos(ciąg, punkt_kontrolny)` –
wyszukuje pierwszy punkt kontrolny w ciągu i
zwraca jego numeryczną pozycję
- `strrpos(ciąg, punkt_kontrolny)` –
identycznie jak `strpos()`, przy czym nie zwraca
uwagi na wielkość znaków

UWAGA: `if (strrpos($ciag, $punkt) !== false) !`

Podstawy PHP – manipulowanie ciągami

- `str_replace(zamieniany, zamiennik, przeszukiwany)` – wyszukuje `zamieniany` ciąg w `przeszukiwanym` i wstawia w jego miejsce `zamiennik`
- `str_ireplace()` – odpowiednik `str_replace()`, nie zwraca uwagi na wielkość liter

Podstawy PHP – komunikaty błędów

- Jak już zostało wspomniane na samym początku, PHP jest bardzo czułe na poprawność składni
- O ile drobne błędy w kodzie HTML przeglądarka jest w stanie "wybaczyć", to serwer WWW najmniejszy błąd opatrzy stosownym komunikatem

Podstawy PHP – komunikaty błędów

- Ogólna postać komunikatu jest następująca:

Typ: Powód

in **x**: \ścieżka\plik.php

on line **Nr_linii**

Podstawy PHP – komunikaty błędów

- Typy komunikatów:
 - Parse error – błąd pierwszej fazy – kompilowania kodu, związany z błędną składnią, powoduje przerwanie wykonywania skryptu

Parse error: syntax error, unexpected 'echo' (T_ECHO) in **D:\xampp\htdocs\php\bledy.php** on line **4** (brak średnika)

Podstawy PHP – komunikaty błędów

- Fatal error – błąd drugiej fazy – wykonywania kodu, związany z odwołaniem do nieistniejącego pliku lub funkcji, powoduje przerwanie wykonywania skryptu

Fatal error: require(): Failed opening required 'funkcje.php' (include_path='\\xampp\\php\\PEAR') in **D:\\xampp\\htdocs\\php\\bledy.php** on line **2**

Podstawy PHP – komunikaty błędów

- Warning – ostrzeżenie, nie powoduje przerwania wykonywania skryptu, zazwyczaj błędne użycie funkcji

Warning: fopen() expects at least 2 parameters, 1 given in **D:\xampp\htdocs\php\bledy.php** on line **2**

Podstawy PHP – komunikaty błędów

- Notice – ma najniższy priorytet, nie powoduje przerwania wykonywania kodu, np. próba wyświetlenia tablicy jak zwykłej zmiennej

Notice: Array to string conversion in

D:\xampp\htdocs\php\bledy.php on line 4

Array

Podstawy PHP – komunikaty błędów

- W pliku `php.ini` mamy możliwość ustawienia poziomu raportowania błędów
- Odpowiedzialna jest za to dyrektywa `error_reporting`, za pomocą której ustawiamy poziom raportowania błędów

Podstawy PHP – komunikaty błędów

- Podstawowe kategorie błędów
 - E_ALL – wszystkie komunikaty
 - E_ERROR – tylko fatal error
 - E_WARNING – tylko ostrzeżenia
 - E_NOTICE – tylko powiadomienia
- Za pomocą znaku ^ możemy wykluczyć kategorię z grupy, np.:

```
error_reporting = E_ALL ^ E_NOTICE
```

Podstawy PHP – bazy danych

- Praca z bazą danych daje o wiele większe możliwości niż z plikami jednorodnymi
- Dzięki wbudowanym mechanizmom nie musimy martwić się o współdzielenie, indeksowanie, czy też wyszukiwanie danych
- Najpopularniejsze serwery baz danych to MySQL, MariaDB, MS SQL Server, PostgreSQL, Oracle

Podstawy PHP – bazy danych

- Do zarządzania bazami MySQL/MariaDB służy bezpłatne narzędzie phpMyAdmin
- Jest ono standardowo dołączane do aplikacji xampp
- phpMyAdmin został napisany w PHP i wymaga włączonego serwera WWW i MySQL/MariaDB

Podstawy PHP – bazy danych

- Dzięki phpMyAdmin za pomocą kilku kliknięć utworzymy bazę danych i dodamy do niej tabelę wraz z kolumnami
- Aby zadbać o bezpieczeństwo, należy utworzyć nowego użytkownika i nadać mu uprawnienia jedynie do zarządzania bazą danych

- Nowa
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test

Bazy danych

Utwórz bazę danych

my_database

utf8_general_ci

Utwórz

Baza danych	Metoda porównywania napisów	Działanie
<input type="checkbox"/> information_schema	utf8_general_ci	Sprawdź uprawnienia
<input type="checkbox"/> mysql	latin1_swedish_ci	Sprawdź uprawnienia
<input type="checkbox"/> performance_schema	utf8_general_ci	Sprawdź uprawnienia
<input type="checkbox"/> phpmyadmin	utf8_bin	Sprawdź uprawnienia
<input type="checkbox"/> test	latin1_swedish_ci	Sprawdź uprawnienia
Ogółem: 5	latin1_swedish_ci	



Ostatnie Ulubione



- Nowa
- information_schema
- mysql
- my_database
- performance_schema
- phpmyadmin
- test

Serwer: 127.0.0.1 » Baza danych: my_database

Struktura SQL Szukaj Zapytanie Eksport Import Operacje Uprawnienia

⚠ Nie znaleziono żadnych tabel w bazie danych.

Utwórz tabelę

Nazwa: forms

Liczba kolumn: 12

Nazwa tabeli: Dodaj column(s) Wykonaj

Nazwa	Typ	Długość/Wartości	Ustawienia domyślne	Metoda porównywania napisów	Atrybuty	Null	Indeks	A
<input type="text" value="id"/>	INT	<input type="text"/>	Brak	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
Pick from Central Columns								
<input type="text" value="name"/>	VARCHAR	50	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
Pick from Central Columns								
<input type="text" value="surname"/>	VARCHAR	100	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
Pick from Central Columns								
<input type="text" value="weight"/>	VARCHAR	5	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
Pick from Central Columns								
<input type="text" value="height"/>	VARCHAR	5	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
Pick from Central Columns								
<input type="text" value="gender"/>	VARCHAR	1	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
Pick from Central Columns								
<input type="text" value="fruits"/>	VARCHAR	255	Brak	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	---	<input type="checkbox"/>

comment

VARCHAR



500

Brak





[Pick from Central Columns](#)

radius

FLOAT



Brak





[Pick from Central Columns](#)

numA

FLOAT



Brak





[Pick from Central Columns](#)

numB

FLOAT



Brak





[Pick from Central Columns](#)



Struktura tabeli



Relation view

	#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo
<input type="checkbox"/>	1	id	int(11)			Nie	Brak		AUTO_INCREMENT
<input type="checkbox"/>	2	name	varchar(50)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	3	surname	varchar(100)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	4	weight	varchar(5)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	5	height	varchar(5)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	6	gender	varchar(1)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	7	fruits	varchar(255)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	8	comment	varchar(500)	utf8_general_ci		Tak	NULL		
<input type="checkbox"/>	9	radius	float			Tak	NULL		
<input type="checkbox"/>	10	numA	float			Tak	NULL		
<input type="checkbox"/>	11	numB	float			Tak	NULL		

Podstawy PHP – bazy danych

- Podstawowymi operacjami na bazach danych jest wstawianie danych oraz ich pobieranie
- Operacje te wykonuje się tworząc zapytania w języku SQL
- Podobnie jak PHP, język SQL jest bardzo restrykcyjny jeśli chodzi o składnię

Podstawy PHP – bazy danych

- Wstawianie danych do bazy realizowane jest za pomocą następującego polecenia:

```
INSERT INTO `nazwa_tabeli`  
(`nazwa_kol1`, `nazwa_kol2`, ...)  
VALUES  
('wart1', 'wart2', ...)
```

UWAGA: możemy podać nazwy tych kolumn, do których chcemy coś wstawić

Podstawy PHP – bazy danych

```
INSERT INTO `nazwa_tabeli` VALUES  
(NULL , 'wart1', 'wart2', ...)
```

UWAGA: jeśli nie podajemy nazw kolumn, to musimy podać wartości dla wszystkich kolumn

Podstawy PHP – bazy danych

```
INSERT INTO `forms`
```

```
(`name`, `surname`, `height`, `weight`) VALUES
```

```
('Jan', 'Kowalski', 1.75, 95)
```

[Pokaż okno zapytań](#)

✓ Wstawionych rekordów: 1. [br] Id wstawionego wiersza: 1 (Wykonanie zapytania trwało 0.0028 sekund(y).)

```
INSERT INTO `forms` (`name`, `surname`, `height`, `weight`) VALUES ('Jan', 'Kowalski', 1.75, 95)
```

✓ Pokazano wiersze 0 - 0 (1 ogółem, Wykonanie zapytania trwało 0.0005 sekund(y).)

```
SELECT * FROM `forms`
```

☐ Profilowanie [[Edit inline](#)] [[Edytuj](#)]

☐ Pokaż wszystko

Liczba wierszy:

25



Filtrowanie wierszy:

+ Opcje



id

name

surname

weight

height

gender

fruits

comment

radius

numA

numB



Edytuj



Kopiuj



Usuń

1

Jan

Kowalski

95

1.75

NULL

NULL

NULL

NULL

NULL

NULL



Zaznacz wszystko

Z zaznaczonymi:



Edytuj



Kopiuj



Usuń



Eksport

Podstawy PHP – bazy danych

```
INSERT INTO `forms`
```

```
(`name`, `surname`, `height`, `weight`) VALUES
```

```
('Paweł', 'Gaweł', 1.95, 95),
```

```
('Adam', 'Abacki', 2, 115),
```

```
('Anna', 'Dziewanna', 1.65, 50)
```

Pokaż okno zapytań

✓ Wstawionych rekordów: 3. [br] Id wstawionej wiersza: 4 (Wykonanie zapytania trwało 0.0055 sekund(y).)

```
INSERT INTO `forms` (`name`, `surname`, `height`, `weight`) VALUES ('Paweł', 'Gaweł', 1.95, 95), ('Adam', 'Abacki', 2, 115), ('Anna', 'Dziewanna', 1.65, 50)
```

[[Edit inline](#)] [[Edytuj](#)] [[Utwórz kod PHP](#)]

✓ Pokazano wiersze 0 - 3 (4 ogółem, Wykonanie zapytania trwało 0.0006 sekund(y).)

```
SELECT * FROM `forms`
```

☐ Profilowanie [[Edit inline](#)] [[Edytuj](#)]

☐ Pokaż wszystko

Liczba wierszy:

25

Filtrowanie wierszy:

Sortuj wg klucza:

Żaden

+ Opcje



id	name	surname	weight	height	gender	fruits	comment	radius	numA	numB
1	Jan	Kowalski	95	1.75	NULL	NULL	NULL	NULL	NULL	NULL
2	Paweł	Gaweł	95	1.95	NULL	NULL	NULL	NULL	NULL	NULL
3	Adam	Abacki	115	2	NULL	NULL	NULL	NULL	NULL	NULL
4	Anna	Dziewanna	50	1.65	NULL	NULL	NULL	NULL	NULL	NULL

Podstawy PHP – bazy danych

- Aby pobrać dane z tabeli, należy użyć komendy SELECT:

```
SELECT nazwa_kol FROM nazwa_tabeli;
```

```
SELECT kol1, kol2 FROM nazwa_tabeli;
```

```
SELECT * from nazwa_tabeli;
```














```
SELECT * from nazwa_tabeli WHERE id = 5;
```

✓ Pokazano wiersze 0 - 3 (4 ogółem, Wykonanie zapytania trwało 0.0007 sekund(y).)

```
SELECT id, name, surname FROM `forms`
```

☐ Pokaż wszystko | Liczba wierszy: 25 Filtrowanie wierszy:

+ Opcje

			id	name	surname
<input type="checkbox"/>	 Edytuj	 Kopiuj	 Usuń	1	Jan Kowalski
<input type="checkbox"/>	 Edytuj	 Kopiuj	 Usuń	2	Paweł Gawęł
<input type="checkbox"/>	 Edytuj	 Kopiuj	 Usuń	3	Adam Abacki
<input type="checkbox"/>	 Edytuj	 Kopiuj	 Usuń	4	Anna Dziewanna



Zaznacz wszystkie

7 zaznaczonych



Edytuj



Kopiuj



Usuń



Widok

Podstawy PHP – bazy danych

- łączenie z bazą danych za pomocą PHP

```
$host = 'localhost';
```

```
$db = 'my_database';
```

```
$user = 'root'; //utworzyć innego  
użytkownika
```

```
$pass = ''; //koniecznie utworzyć  
hasło
```

Podstawy PHP – bazy danych

- Najbezpieczniejszym rozwiązaniem jest utworzenie oddzielnego pliku z danymi dostępowymi do bazy oraz zapisanie go poza folderem publicznym
- Następnie plik należy dołączyć za pomocą funkcji `include()` lub `require()`;

Podstawy PHP – bazy danych

```
$public = $_SERVER['DOCUMENT_ROOT'];  
require_once("$public/../../include/db.php"); //jeśli nie znajdzie pliku,  
to fatal error, sprawdza czy już wcześniej nie został dołączony  
  
$conn = new mysqli($host, $user, $pass, $db);  
if ($conn->connect_errno){  
    echo 'Brak połączenia z bazą danych: ' . $conn->connect_error;  
} else {  
    echo 'Połączono.';  
}  
$conn->set_charset("utf8");
```

Podstawy PHP – bazy danych

- Zapytanie do bazy danych:

```
$query = 'SELECT * FROM forms';  
$rows = $conn->query($query);  
if (!$rows) {  
    echo 'Brak wyników dla  
zapytania.';  
}
```


Podstawy PHP – bazy danych

```
//Liczba wyników
$num = $rows->num_rows;
//Wydobycie danych z obiektu
for ($i=0; $i < $num; $i++) {
    $row = $rows->fetch_assoc();
    foreach ($row as $k => $v) {
        echo "$k => $v<br>";
    }
}
```

Podstawy PHP – bazy danych

```
//usunięcie danych ze zmiennej  
$rows->free();  
  
//zamknięcie połączenia z bazą  
danych  
$conn->close();
```

Podstawy PHP – bazy danych

Zadanie 6 PHP.pdf

Pytania

?

Dziękuję za uwagę!