Temat: Wprowadzenie do języka PHP

Zespół Szkół Komunikacji ; Gabriela Ledachowicz

Kryteria:

1. Wyjaśniam czym jest PHP, jak działa w sieci, na jakiej licencji mogę z niego korzystać i do czego stosować

PHP – co to takiego?

PHP (rekurencyjny skrót dla *PHP: Hypertext Preprocessor*) jest szeroko stosowanym językiem skryptowym ogólnego zastosowania open-source, który jest szczególnie przydatny do tworzenia stron internetowych i może być osadzony w HTML.

PHP – co to takiego?

- ✓ język programowania przeznaczony do tworzenia i generowania dynamicznych stron WWW
- ✓ jest językiem *interpretowanym* (co oznacza, że nie musisz programu kompilować przed wykonaniem)
- ✓ obecnie można w nim pisać także zwyczajne aplikacje dla systemu operacyjnego

PHP - Licencja

- ✓ rozpowszechniany na zasadach open source
- ✓ można pobrać za darmo jego kopię, zainstalować i używać bez żadnych ograniczeń zarówno do celów prywatnych jak i komercyjnych
- ✓ dostępny jest pełen kod źródłowy

PHP - wykorzystanie

- ✓ dynamiczne generowanie zawartości strony internetowej
- ✓ tworzenie i edytowanie plików na serwerze
- ✓ przetwarzanie danych z formularzy w witrynach internetowych,
- ✓ zaawansowane przetwarzanie tekstu,
- ✓ odbieranie i wysyłanie plików cookies
- ✓ sterowanie dostępem użytkowników do stron witryny internetowej
- ✓ szyfrowanie danych

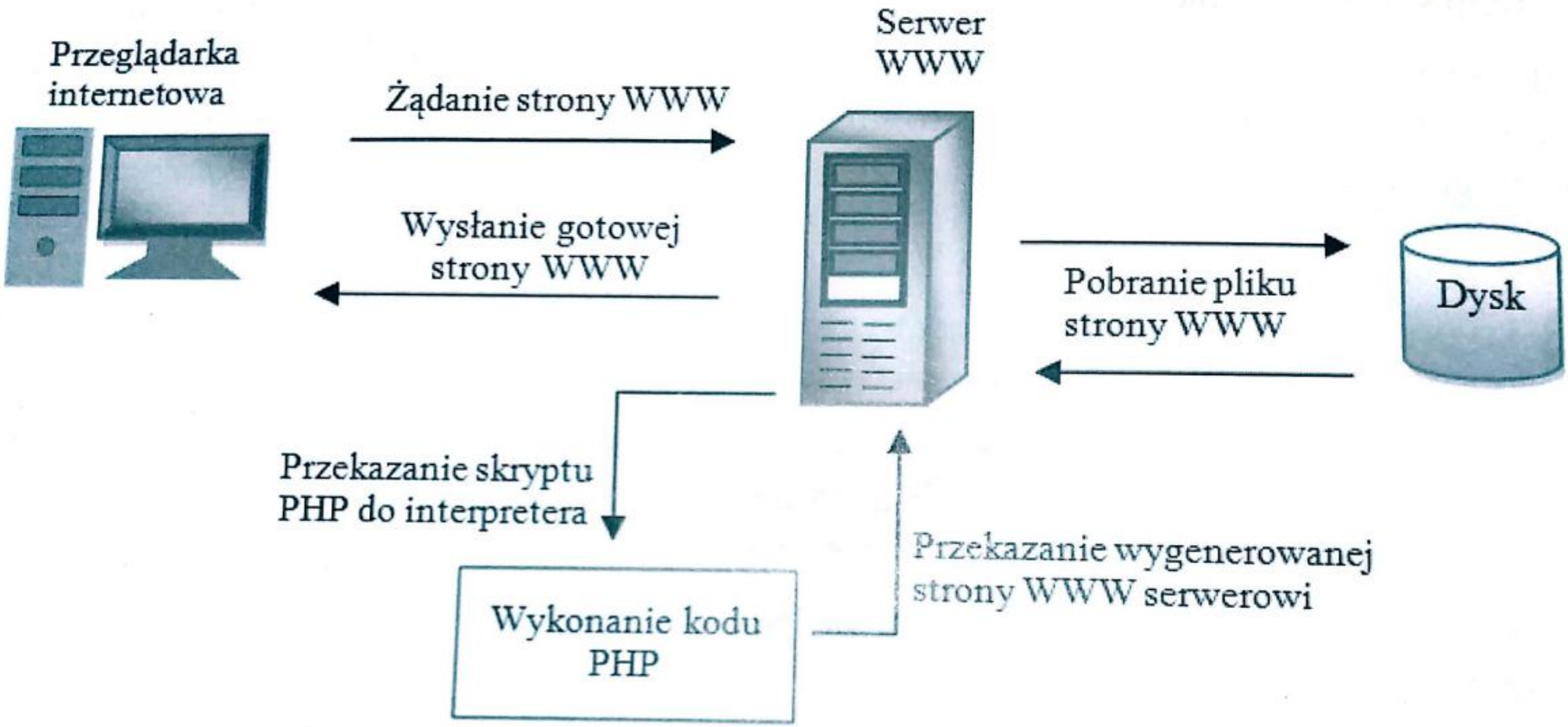
PHP - wykorzystanie

- ✓tworzenie skomplikowanych projektów umożliwiających współpracę z dużymi bazami danych.
- ✓obsługuje wiele protokołów sieciowych, takich jak NNTP, SMTP, POP3 czy IMAP,
- ✓pozwala na dynamiczne tworzenie obrazów, dokumentów PDF czy danych XML.

PHP

- ✓ PHP jest językiem server-side, tj. **pracuje po stronie serwera WWW**
- ✓ Przeciwieństwem są języki client-side pracujące po stronie przeglądarki użytkownika (np. JavaScript)
- ✓Język skryptowy

Proces przetwarzania kodu PHP



Kryteria:

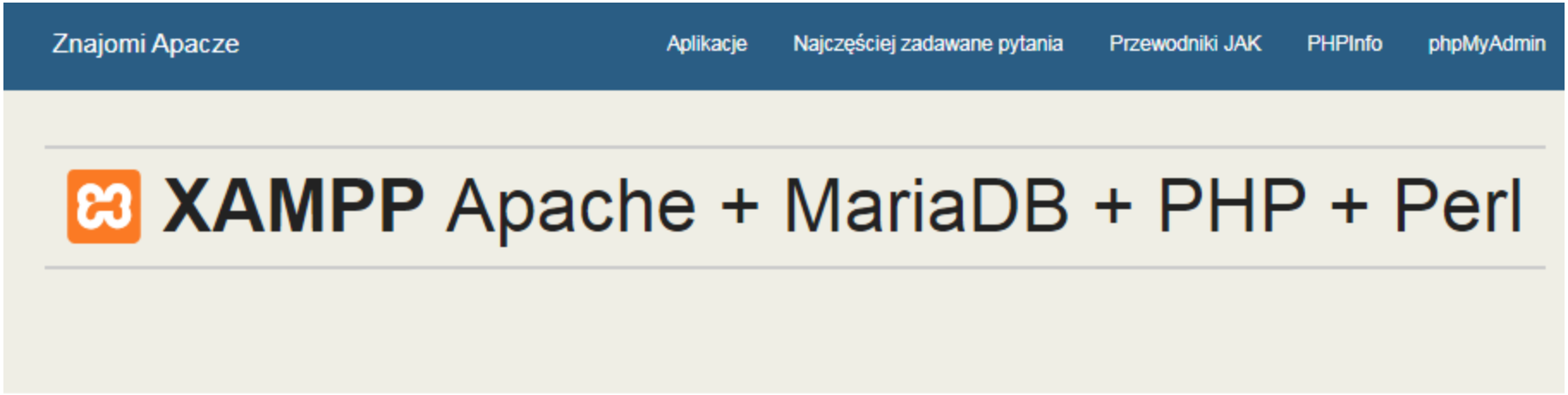
1. Wyjaśniam czym jest PHP, jak działa w sieci, na jakiej licencji mogę z niego korzystać i do czego stosować

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium

Kryteria:

2. Wyjaśniam gdzie i w jaki sposób umieścić kod PHP w dokumencie

PHP – XAMPP


Znajomi Apacze
Aplikacje
Najczęściej zadawane pytania
Przewodniki JAK
PHPInfo
phpMyAdmin
XAMPP Apache + MariaDB + PHP + Perl
Witamy w XAMPP dla Windows 7.2.9
Pomyślnie zainstalowałeś XAMPP w tym systemie! Teraz możesz zacząć używać Apache, MariaDB, PHP i innych składników. Więcej informacji można znaleźć w sekcji często zadawanych pytań lub zapoznać się z przewodnikiem JAK, aby rozpocząć korzystanie z aplikacji PHP.
XAMPP jest przeznaczony tylko do celów programistycznych. Ma pewne ustawienia konfiguracyjne, które ułatwiają rozwój lokalny, ale są niepewne, jeśli chcesz, aby twoja instalacja była dostępna dla innych. Jeśli chcesz, aby Twój XAMPP był dostępny z internetu, upewnij się, że rozumiesz jego konsekwencje i sprawdziłeś często zadawane pytania, aby dowiedzieć się, jak chronić swoją witrynę. Alternatywnie możesz użyć WAMP , MAMP lub LAMP, które są podobnymi opakowaniami, które są bardziej odpowiednie do produkcji.

PHP – phpinfo()

PHP Version 7.2.9

System	Windows NT RAINELOU 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Aug 15 2018 23:06:47
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp1\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718

PHP – phpinfo()

SERVER_NAME	localhost
DOCUMENT_ROOT	C:/xampp1/htdocs
PHP Version	7.2.9
Loaded Configuration File	C:\xampp1\php\php.ini
max_file_uploads	20

PHP - Umieszczanie kodu PHP w dokumencie

Skrypty PHP są najczęściej wykonywane przez serwer WWW. W celu wyodrębnienia kodu skryptu od innych elementów (np. kodu HTML) niezbędne jest **umieszczenie go wewnątrz znaczników.**

Przetwarzane są jedynie fragmenty kodu, które znajdą się pomiędzy znacznikiem otwierającym i zamykającym kod.

PHP - Umieszczanie kodu PHP w dokumencie

Interpeter PHP analizuje plik, szuka tagów otwierających i zamykających skrypt. PHP interpretuje kod osadzony między nimi.

Parsowanie w ten sposób pozwala na osadzenie PHP w różnego rodzaju dokumentach, ponieważ parowanie tagów otwierających i zamykających ignoruje wszystko poza parą.

PHP7 – znaczniki kanoniczne

Początek i koniec programu mogą być oznaczone odpowiednio przez:

```php
<?php
//kod skryptu
?>
```

Najczęściej spotykane, zawsze rozpoznawane niezależnie od tego, jakie opcje włączy się w pliku konfiguracyjnym. Zaleca się stosowanie tych znaczników

PHP7 – znaczniki kanoniczne

Jeżeli plik zawiera jedynie kod skryptu można pominąć znacznik zamykający:

```
<?php
//kod skryptu
```

Dzięki temu poza trybem PHP nie pozostaną żadne niepotrzebne białe znaki (spacje, entery, tabulatory)

PHP7 – typu SGML

Znaczniki typu SGML mają postać:

```
<?
//kod skryptu
?>
```

Najkrótsza forma znaczników PHP.
Aby ich używać należy umieścić w pliku konfiguracyjnym *php.ini* linię: **`short_open_tag=On`**
Nie należy ich stosować w przypadku zagnieżdżania kodu PHP w plikach XML i XHTML

Kryteria:

2. Wyjaśniam gdzie i w jaki sposób umieścić kod PHP w dokumencie

---

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium

Kryteria:

3. Podaję sposoby zamieszczenia komentarzy w kodzie PHP

PHP – Komentarze

Dobry zwyczaj nakazuje komentować pisany kod. Komentarz wewnątrz pary znaczników może być oznaczany na 3 sposoby.

Komentarz blokowy:

```
<?php
/*
to jest komentarz blokowy
*/
?>
```

PHP – Komentarze

Komentarz jednowierszowy:

```php
<?php
// komentarz jednowierszowy
?>
```

Rozpoczyna się od znaków // i obowiązuje do końca wiersza

PHP – Komentarze

Komentarz jednowierszowy uniksowy:

```php
<?php
# komentarz jednowierszowy
?>
```

Rozpoczyna się od znaków # i obowiązuje do końca wiersza

Kryteria:

3. Podaję sposoby zamieszczenia komentarzy w kodzie PHP

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium

Temat: Wyświetlanie informacji, łączenie skryptów

Zespół Szkół Komunikacji ; Gabriela Ledachowicz

Kryteria:

4. Wymieniam instrukcje PHP pozwalające wyświetlać tekst na stronie oraz sposoby osadzania skryptu w dokumencie

PHP – zasady

- ✓W plikach z rozszerzeniem php można używać zarówno kodu php jak i HTML.
- ✓**Każda instrukcja w php kończy się średnikiem!**
- ✓Instrukcje mogą być pisane pojedynczo w jednym wierszu lub w kilku .
- ✓Znacznik zamykający bloku kodu PHP automatycznie oznacza średnik; nie trzeba wstawiać średnika kończącego ostatnią linię bloku PHP.

PHP – Wyświetlanie tekstu na stronie

Instrukcje pozwalające na wypisanie tekstu:

print	Najprostsza funkcja wypisująca podaną wartość. Może być wykorzystywana w bardziej złożonych wyrażeniach
echo	Instrukcja wypisująca podaną wartość, podobna do print.
printf	Funkcja pozwalająca na wypisanie wartości po jej wcześniejszym sformatowaniu.

instrukcje **echo** i **print** mogą być używane z lub bez nawiasów: echo lub echo (), print lub print()

PHP – Wyświetlanie na stronie

Instrukcja echo

```
echo ("ciąg_znaków");          echo ('ciąg_znaków');
echo "ciąg_znaków";            echo 'ciąg_znaków';
```

Aby wyświetlić **wartość liczbową**, nie trzeba jej ujmować w znaki cudzysłowu, można bowiem zastosować taką konstrukcję:

```
echo wartość;
Np.   echo 25;
```

PHP – Wyświetlanie na stronie

Istnieje możliwość wysłania za pomocą instrukcji echo wielu wartości, zarówno liczbowych, jak i ciągów znaków, które w takim przypadku należy oddzielić od siebie znakami przecinka:

echo wartość1, wartość2, ..., wartośćN
np.:
echo "Liczba ", 24, " jest", " dwucyfrowa.";

Nie można jednak ujmować wszystkich wartości w zbiorczy nawias okrągły

PHP – Wyświetlanie na stronie

Instrukcja print

W przypadku **print** nie można wyświetlanych wartości oddzielać od siebie znakami przecinka

Zabroniona jest zatem konstrukcja:

```
print "Liczba ", 24, " jest", " dwucyfrowa.";
```

Prawidłowe będą natomiast zapisy:

```
print "Wartość";
print 24;
print ("Słowo");
print (24);
print "<h2 style=\"color:blue\">";
```

PHP – osadzanie skryptu na stronie

```php
<?php
    // instrukcja echo wyświetla ciąg znaków
    echo "Cześć!";
?>
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <meta charset="utf-8"/>
  <title>Mój pierwszy program PHP</title>
</head>
<body>
</body>
</html>
```

http://localhost/zsk/l2p1.php

PHP – osadzanie skryptu na stronie

```
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4    <meta charset="utf-8"/>
5    <title>Mój pierwszy program PHP</title>
6  </head>
7  <body>
8      <h1>
9          <?php
10          echo "Cześć!";
11         ?>
12     </h1>
13 </body>
14 </html>
```

http://localhost/zsk/l2p2.php

PHP – osadzanie skryptu na stronie

```
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4      <meta charset="utf-8"/>
5      <title>Mój pierwszy program PHP</title>
6  </head>
7  <body>
8      <?php
9          echo "<h1> Cześć! </h1>";
10     ?>
11     </body>
12 </html>
```

http://localhost/zsk/l2p3.php

Kryteria:

4. Wymieniam instrukcje PHP pozwalające wyświetlać tekst na stronie oraz sposoby osadzania skryptu w dokumencie

---

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium

Kryteria:

5. Określam metody definiowania stringów oraz bloków tekstowych oraz wskazuję na możliwe błędy przy ich stosowaniu

PHP – metody definiowania stringów oraz bloków tekstowych

Napisy w języku PHP definiujemy na trzy sposoby:

1. stosując **apostrofy '**

```
echo 'Ala';
```

2. stosując **cudzysłów "**

```
echo "Ala";
```

3. stosując **notację *heredoc* *lub nowdoc***

```
echo <<<EOD
Ala
EOD;
```

```
echo <<<'EOD'
Ala
EOD;
```

PHP – użycie cudzysłowu lub apostrofu

Problem - gdy znacznik musi zawierać argument wymagający ujmowania wartości w znaki apostrofu lub cudzysłowu.

Np. <h2 style="color:blue">

W takiej sytuacji przed znakami cudzysłowów występującymi wewnątrz ciągu można wstawić znaki ukośnika:

```php
<?php
echo "<h2 style=\"color:blue\">";
?>
```

PHP – użycie cudzysłowu lub apostrofu

Lub zastosować apostrof, który jest „silniejszy” od cudzysłowu. Cudzysłów może być wykorzystany pomiędzy apostrofami, nie myląc interpretera.

Np.

```php
<?php
    echo '<img src= "obraz.jpg"><br>';
?>
```

PHP – heredoc

- ✓ Heredoc (here-document) umożliwia wstawianie bloków tekstowych z zachowaniem miejsc łamania wierszy i innych białych znaków (w tym wcięć)
- ✓ Przydaje się przy pisaniu długiego kodu z poziomu PHP bez potrzeby ciągłego wychodzenia z niego
- ✓ pozwala zdefiniować własny ogranicznik ciągu znaków,

PHP – duży kod, składnia

```
<body>
    <h1><?php echo 'Wypisanie dużego tekstu' ?></h1>
    <?php
        echo 'Dłuższy kod <br>';
        echo 'warto zapisać<br>';
        echo 'przy pomocy składni heredoc<br>';
        echo '<p>';
        echo 'Cztery punkty i siedem lat temu <br>';
        echo 'nasi ojcowie wkroczyli na ten kontynent.<br>';
        echo '(i tak dalej ...) <br>';
        echo '</p>';
    ?>
    </body>
```

http://localhost/zsk/l2p5.php

PHP – heredoc

```php
<?php
echo<<<KONIEC
<h1>Wypisanie dużego tekstu</h1>
Dłuższy kod <br>
warto zapisać<br>
przy pomocy składni heredoc<br>
<p>
Cztery punkty i siedem lat temu <br>
nasi ojcowie wkroczyli na ten
kontynent.<br>
(i tak dalej ...) <br>
</p>
KONIEC;
?>
```

http://localhost/zsk/l2p6.php

http://localhost/zsk/l2p6a.php

PHP – heredoc

- ✓łańcuch znakowy należy rozpocząć od sekwencji <<<,
- ✓po niej musi nastąpić identyfikator,
- ✓tego identyfikatora należy następnie użyć w celu zasygnalizowania końca łańcucha znakowego,
- ✓nazwa identyfikatora może się zaczynać wyłącznie od znaku podkreślenia lub litery i może zawierać dowolną kombinację liter, cyfr i znaków podkreślenia.
- ✓linia kończąca nie może natomiast zawierać żadnych innych znaków oprócz identyfikatora i średnika.
- ✓Uwaga ta dotyczy wszystkich znaków, również spacji, tabulatorów itp.!!

Kryteria:

5. Określam metody definiowania stringów oraz bloków tekstowych oraz wskazuję na możliwe błędy przy ich stosowaniu

---

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium

Kryteria:

6. Wyjaśniam działanie instrukcji służących do łączenia skryptów i różnicę między nimi

PHP – Łączenie skryptów

Skrypty PHP mogą zawierać dużą ilość kodu i być bardzo skomplikowane.
W takiej sytuacji najczęściej kod jest dzielony i zapisywany w osobnych plikach, którymi łatwiej zarządzać i które łatwiej analizować.
Podzielone pliki można połączyć przy pomocy instrukcji **include i require.**

PHP – Łączenie skryptów

Instrukcja include

include("nazwa_pliku"); include "nazwa_pliku";

```
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4      <meta charset="utf-8"/>
5      <title>Mój pierwszy program PHP</title>
6  </head>
7  <body>
8      <?php
9          include "l2p7skrypt.php";
10     ?>
11     </body>
12 </html>
```

http://localhost/zsk/l2p7.php

PHP – Łączenie skryptów

Instrukcja include

```php
<?php
echo "<h2 style='text-align:center'>";
echo "Witamy na stronie!";
echo "</h2>";
?>
```

http://localhost/zsk/l2p7skrypt.php

PHP – Łączenie skryptów

Instrukcja include

instrukcja include o schematycznej postaci:

include "nazwa_pliku";

jest w rzeczywistości traktowana jako konstrukcja:

?>

treść pliku nazwa_pliku

<?php

Jeśli chcemy, aby treść dołączanego pliku była traktowana jako kod PHP, musimy koniecznie ująć ją w znaczniki trybu PHP

http://localhost/zsk/l2p7skrypt.php

PHP – Łączenie skryptów

Instrukcja require

✓ma działanie bardzo podobne do include— wczytuje plik zewnętrzny o wskazanej nazwie.
✓różnica ujawnia się w momencie, gdy wybrany plik nie może zostać wczytany (np. nie ma go na dysku lub PHP nie ma do niego praw dostępu).
✓w przypadku instrukcji **include** wygenerowane zostanie ostrzeżenie, ale skrypt, w którym zawarte było jej wywołanie, będzie kontynuował działanie.
✓w przypadku **require** skrypt wywołujący zakończy działanie, zgłaszając błąd

PHP – Łączenie skryptów

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Moja strona WWW</title>
6  </head>
7  <body>
8  <div>
9  <?php
10 echo("Przed wywołaniem include...<br>");
11 include "abc.php";
12 echo("Po wywołaniu include...<br>");
13 echo("Przed wywołaniem require...<br>");
14 require "abc.php";
15 echo("Po wywołaniu require...<br>");
16 ?>
17 </div>
18 </body>
19 </html>
```

http://localhost/zsk/l2p8.php

Kryteria:

6. Wyjaśniam działanie instrukcji służących do łączenia skryptów i różnicę między nimi

---

Pokaż światłami/kciukami jak oceniasz, na ile spełniasz kryterium