

PHP  
1.21

Learn

+

← → ↺ 🏠

🔒 https://hourofcode.com/pl/learn

☆ 🔍 Szukaj

🔖 📖 ☰



Stwórz grę Flapp

Klasy 2+ | Bloki



Kodable (zaprasza

Od dzieci, które uczą się nowoczesne przeglądarki



CODE COM



**Napisz swój pierwszy program komputerowy**

Code.org

Klasy 2+ | Bloki

Poznaj podstawowe pojęcia informatyki, przeciągając i upuszczając bloki z kodem. Jest to samouczek w stylu gry, wykorzystujący filmiki z udziałem Billa Gatesa, Marka Zuckerberga, Angry Birds i Plants vs. Zombies. Poznaj pętle "powtarzaj", warunki oraz podstawowe algorytmy. Dostępne w 37 językach.

Rozpocznij

|                      |                                                                                                                                                                                                                                                                                     |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Więcej zasobów       | 🔗 <b>Notatki nauczyciela</b>                                                                                                                                                                                                                                                        |
| Krótki link          | <a href="https://hourofcode.com/code">https://hourofcode.com/code</a>                                                                                                                                                                                                               |
| Przygotowanie ucznia | Początkujący                                                                                                                                                                                                                                                                        |
| Technologia w klasie | Wszystkie współczesne przeglądarki, Android, iOS                                                                                                                                                                                                                                    |
| Tematy               | Informatyka tylko                                                                                                                                                                                                                                                                   |
| Rodzaj aktywności    | Samouczek                                                                                                                                                                                                                                                                           |
| Długość              | Jedna godzina, Godzina z kontynuacją                                                                                                                                                                                                                                                |
| Języki               | Angielski, arabski, bułgarski, chiński, chorwacki, duński, holenderski, fiński, francuski, niemiecki, grecki, islandzki, indonezyjski, włoski, japoński, koreański, norweski, perski, polski, portugalski, rumuński, rosyjski, hiszpański, szwedzki, turecki, ukraiński i 19 więcej |



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS



move forward by 100 pixels

turn right by 90 degrees



ANGRY BIRDS



ANGRY BIRDS

Scratch - Imagine, Program, Sho X

+

← → ↺ 🏠

🔒 https://scratch.mit.edu

☆ 🔍 Szukaj

🔒 ||| ☰

 Stwórz Przeglądaj Pomysły Info 🔍 Szukaj

Dołącz do Scratch Zaloguj się

🌟 Scratch to największa na świecie społeczność dzieci - programistów. Twoje wsparcie ma znaczenie. 

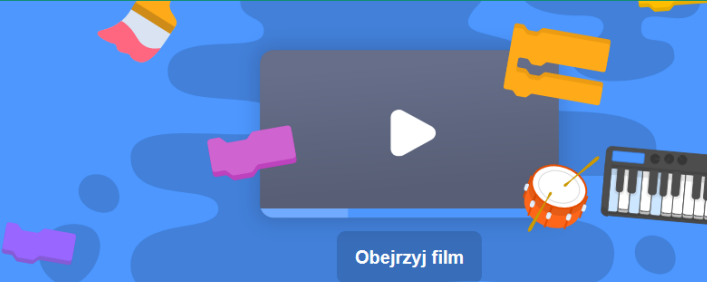
Prześlij darowiznę

# Twórz historyjki, gry i animacje

## Dziel się z ludźmi na całym świecie

🌟 Rozpocznij tworzenie

⚡ Dołącz



Obejrzyj film

O Scratchu

Dla Rodziców

Dla nauczycieli

### Wyróżnione projekty



DYO Potato  
katak5786



Stop Motion  
Artsygirlforlife



Flight Cycle  
prettyGirl\_1



stickman barriers  
gustavinhodox1



This is Halloween ...  
-WatermelonGum-

https://scratch.mit.edu/

PHP: Preface - Manual

https://www.php.net/manual/en/preface.php

130% Szukaj

php Downloads Documentation Get Involved Help php

Search

PHP Manual > PHP Manual

« PHP Manual Getting Started »

PHP Manual

» Preface

Change language: English

Submit a Pull Request Report a Bug

## Preface

PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

This manual consists primarily of a [function reference](#), but also contains a [language reference](#), explanations of some of PHP's major [features](#), and other [supplemental](#) information.

You can download this manual in several formats at » <https://www.php.net/download-docs.php>. More information about how this manual is developed can be found in the '[About the manual](#)' appendix. If you are interested in the [history of PHP](#), visit the relevant appendix.



Tutorials ▾

References ▾

Exercises ▾

Videos **NEW**

Website **NEW**

Paid Courses

Log in

## PHP Tutorial

### PHP HOME

PHP Intro  
PHP Install  
PHP Syntax  
PHP Comments  
PHP Variables  
PHP Echo / Print  
PHP Data Types  
PHP Strings  
PHP Numbers  
PHP Math  
PHP Constants  
PHP Operators  
PHP If...Else...Elseif  
PHP Switch  
PHP Loops  
PHP Functions  
PHP Arrays  
PHP Superglobals  
PHP RegEx

### PHP Forms

PHP Form Handling

ADVERTISEMENT

<https://www.w3schools.com/php/>

# PHP Tutorial

< Home

Next >

## Learn PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

[Start learning PHP now »](#)

## Easy Learning with "PHP Tryit"

kurs



## Intro to PHP: Create an Online Store from Scratch

HTML Academy

Grades 6+ | PHP

Meet Dumpo the Little Elephant, a senior programmer at a web design studio, who is developing an online store. Together you will learn the basics of PHP in an engaging way. You will work with web scripts and learn the variables, add PHP-scripts into the layout and interact with databases while creating a true-to-life web project.

Start

|                      |                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------|
| Short link           | <a href="https://hourofcode.com/htmlacademyphp">https://hourofcode.com/htmlacademyphp</a> |
| Student experience   | Comfortable                                                                               |
| Classroom technology | All modern browsers, Screen reader                                                        |
| Topics               | Computer Science only                                                                     |
| Activity type        | Self-led tutorial                                                                         |
| Length               | One hour                                                                                  |
| Languages            | English                                                                                   |

- 1) gathering data 📊
- 2) cleaning the data 🧹
- 3) training the model 🧠
- 4) making predictions 📈

MAKE SCHOOL

with Machine Learn...

on



endent





# PHP w działaniu

Podaj swoje imię

imię:

```
<!doctype html>
<html>
<head>
  <title>pole tekstowe</title>
  <meta charset="utf-8">
</head>
<body>
  <form name="pole_tekstowe" action="imie.php" method="post">
    <p style="font-weight:bold;">Podaj swoje imię</p>
    imię: <input type="text" name="imie" value="tu wpisz swoje imię" size="30">
    <br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

index.html

formularz html

Cześć Marek!

Co słychać?



```
<!doctype html>
<html>
<head>
  <title>imie</title>
  <meta charset="utf-8">
</head>
<body>

<p>Cześć Marek!</p><p>Co słychać?</p>
</body>
</html>
```

```
<!doctype html>
<html>
<head>
  <title>imie</title>
  <meta charset="utf-8">
</head>
<body>

<?php
// deklarujemy zmienną lokalną imie
// inicjalizujemy ja danymi z formularza
$imie=$_POST["imie"];

// przetwarzamy dane i generujemy kod html
echo "<p>Cześć $imie!</p>";
echo "<p>Co słychać?</p>";
?>

</body>
</html>
```

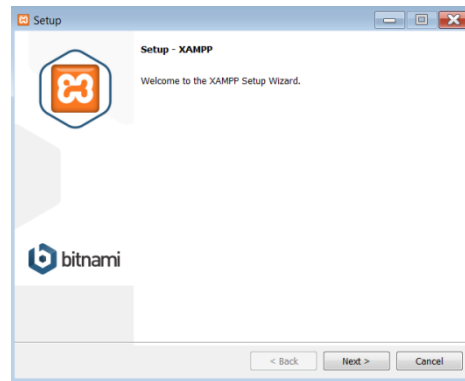
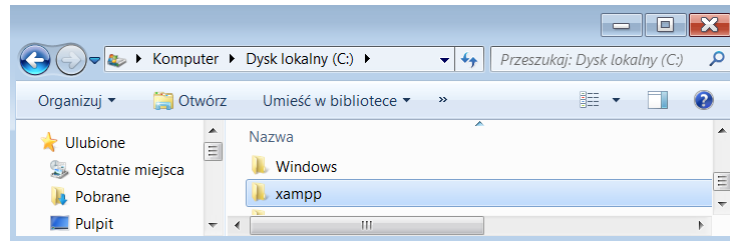
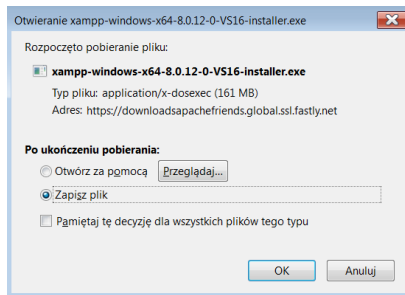
imie.php

# XAMPP instalacja

Instalujemy - domyślnie na dysku C:

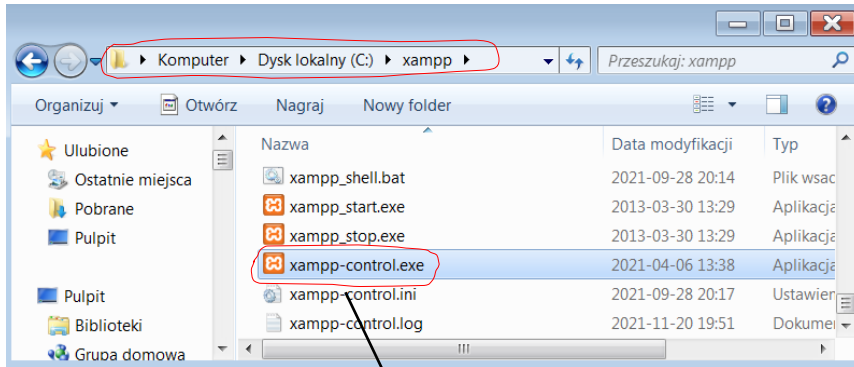


pobieramy



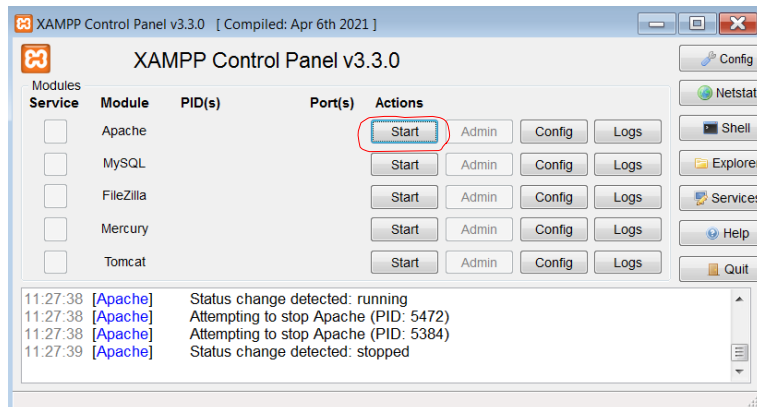
# XAMPP uruchamianie

w pasku adresu przeglądarki wpisujemy localhost



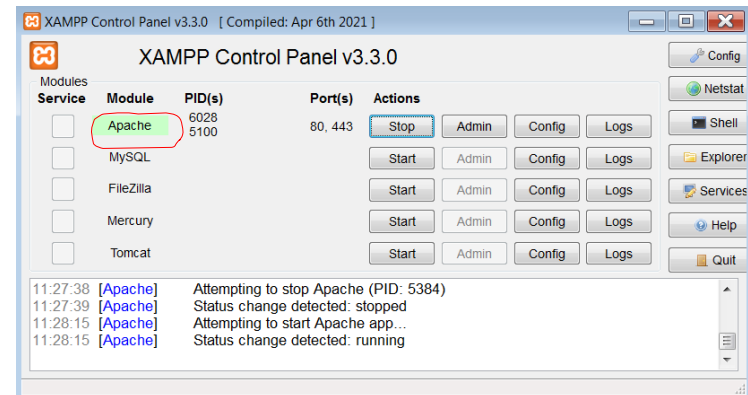
pojawia się strona startowa serwera Apache

uruchamiamy

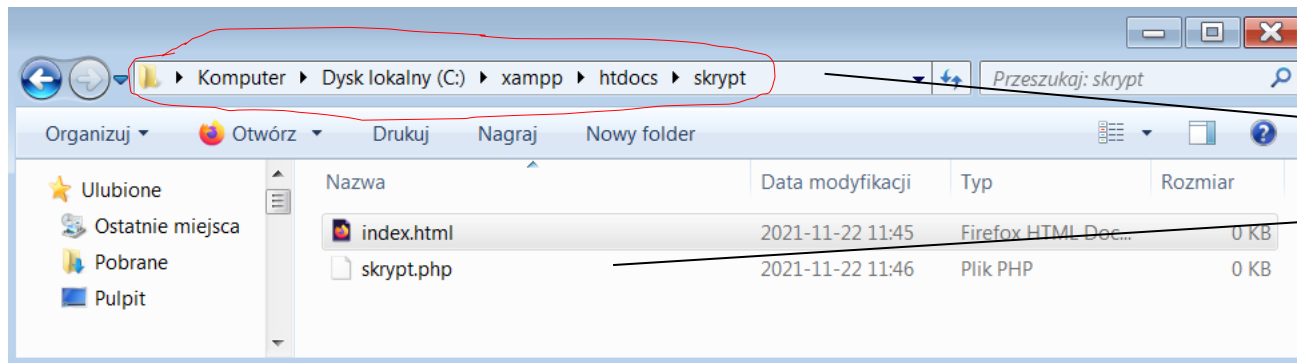


startujemy  
serwer  
Apache

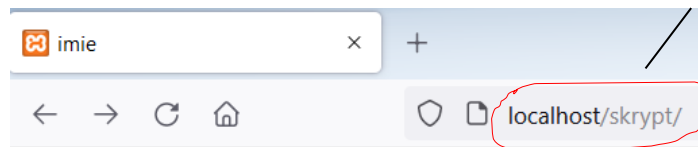
uruchamiamy przeglądarkę



# XAMPP uruchomienie skryptu php



w katalogu C:\xampp\htdocs\ tworzymy własny katalog skrypt oraz umieszczamy w nim pliki index.html i skrypt.php (patrz slajd „PHP w działaniu”)



w przeglądarce w polu adresu wpisujemy: **localhost/skrypt/**

nazwa katalogu z naszym skryptem

Podaj swoje imię:

Marek

submit

strona index.html



Cześć Marek!

Co słyhać?

Po naciśnięciu przycisku submit wyświetla się strona wygenerowana przez skrypt.php

# HTML formularze

## Ankieta

---

|                      |          |
|----------------------|----------|
| <input type="text"/> | imię     |
| <input type="text"/> | nazwisko |

`<input type="text" name="imie">imie<br>`

### pleć

mężczyzna ☐ kobieta ☐

— `mężczyzna<input type="radio" name="plec" value="mężczyzna">`

### zainteresowania

sport ☐ film ☐ elektronika ☐ informatyka ☐

— `sport<input type="checkbox" name="sport" value="1">`

### szkoła

Podstawowa ▾

`<select name="szkola">  
<option value="podstawowa">Podstawowa</option>  
</select>`

### krótko o sobie

|             |
|-------------|
| <div></div> |
|-------------|

`<textarea name="o_sobie" rows=5 cols=40></textarea>`

`<input type="submit" value="wyslij dane na serwer">`

`<input type="reset" value="resetuj ankietę">`

# HTML formularze – isset(), empty()

formularz nie przesyła zmiennych

## Przykład pół wyboru

możesz wybrać kilka opcji:

co lubisz najbardziej?

☐ jeść

☐ spać

☐ bawić się

żadne pole  
nie jest zaznaczone

```
<form action="pola_wyboru.php" method="post" >

<h3>co lubisz najbardziej?</h3>
<p><input type="checkbox" name="a" value="jesc" checked >jeść</p>
<p><input type="checkbox" name="b" value="spac">spać</p>
<p><input type="checkbox" name="c" value="bawic_sie">bawić się</p>
<hr>

<input type="submit" value="wyślij">
<input type="reset" >

</form>
```

```
<?php
// jeśli nie jest zaznaczone żadne pole nie tworzą się zmienne w tablicy $_POST[]
if (!(isset($_POST["a"]) || isset($_POST["b"]) || isset($_POST["c"]))) {
    echo ("nic nie zaznaczyłeś!");
} else {
    if (!empty($_POST["a"]) && $_POST["a"] == "jesc") {
        echo ("Ja też lubię jeść :)<br>");
    }
    if (!empty($_POST["b"]) && $_POST["b"] == "spac") {
        echo ("Dobranoc!!<br>");
    }
    if (!empty($_POST["c"]) && $_POST["c"] == "bawic_sie") {
        echo ("Milej zabawy z PHP!!<br>");
    }
}
?>
```

isset(\$\_POST["c"]) sprawdza czy jest utworzona zmienna pola opcji „bawić się” (name=„c”),  
Jeśli nie jest funkcja zwraca false -  
nie tworzy się zmienna w tablicy \$\_POST[„c”]

funkcja empty(x) sprawdza czy dana zmienna x jest pusta, jeśli jest pusta zwraca true



nic nie zaznaczyłeś!

# skrypt php w dokumencie html

plik index.php

znacznik otwierający

znacznik zamykający

```
<html>
<body>

<?php

$imie="Marek";
echo "Cześć $imie!"

?>

</body>
</html>
```

blok instrukcji

# komentarze

komentarz blokowy

komentarz jednowierszowy

komentarz jednowierszowy  
uniksowy

```
<html>
<body>

<?php
/*
    Powitanie Marka
*/

// zmienna imie
$imie="Marek";

# wypisujemy powitanie
echo "<h1>Cześć $imie!</h1>"

?>

</body>
</html>
```

# Zmienne

\$warzywa=„pomidory”

\$chleb=„chleb\_wiejski”

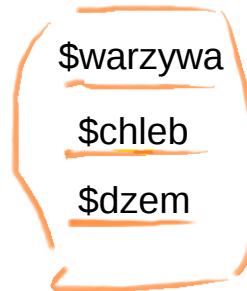
\$dzem=„dzem\_sliwkowy”



zmienne zainicjowane



dane

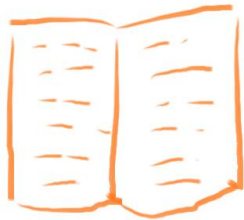


zmienne

/  
lodówka

# Program

Książka Kucharska



algorytmy

program w komputerze



Tata przygotowuje posiłek wg. Książki Kucharskiej

Lodówka



zmienne

\$warzywa=„pomidory”

\$chleb=„chleb\_wiejski”

\$dzem=„dzem\_sliwkowy”

wynik programu



kanapka dla dziecka

```
<!DOCTYPE html>
```

# Program //skrypt php w dokumencie html

```
<html>
```

```
    <title>kanapka</title>
```

```
    <meta charset="utf-8">
```

```
<body>
```

```
<?php
```

```
//Lodówka
```

```
$warzywa="pomidory";
```

```
$chleb="chleb_wiejski";
```

```
$dzem="dzem_sliwkowy";
```

```
//przygotowanie kanapki z Ksiazki Kucharskiej
```

```
$kanapka = "$chleb z $dzem";
```

```
//serwujemy kanapkę dziecku
```

```
echo "Prosze kanapkę: $kanapka oraz $warzywa do smaku, smacznego!";
```

```
?>
```

```
</body>
```

```
</html>
```

# Zmienne

wielkość liter ma znaczenie!

nazwa zmiennej  
( w tym przypadku "a")  
jest poprzedzona znakiem \$

może być deklarowana  
w dowolnym miejscu  
w skrypcie

\$a

poprawna nazwa zmiennej  
zaczyna się od litery  
lub znaku podkreślenia "\_",  
po których może wystąpić  
dowolna ilość liter,  
cyfr lub znaków podkreślenia.

zmienna a

słabe typowanie –  
typ danych jest ustalany  
w czasie działania programu

# Typy danych

- **String** (ciąg znaków)
- **Integer** (liczba całkowita)
- **Float (floating point numbers - also called double)** (liczba zmiennoprzecinkowa, rzeczywista)
- **Boolean** (typ logiczny)
- **Array** (typ tablicowy)
- **Object** (typ obiektowy)
- **NULL** (typ pusty)
- **Resource** (odwołanie do zasobu)

# Zmienne

```
<?php
$i = 5;
echo var_dump($i); //int(5)
echo "<br>";

$f = 5.5;
echo var_dump($f); //float(5.5)
echo "<br>";

$s = '5';
echo var_dump($s); //string(1) "5"
echo "<br>";

$s1 = "55";
echo var_dump($s1); //string(2) "55"
echo "<br>";

$b = true;
echo var_dump($b); //bool(true)
echo "<br>";
?>
```

słabe typowanie –  
nie musimy podawać typu  
przy inicjalizacji zmiennej,  
interpreter zrobi to za nas

# Zmienne - deklaracje typu

deklaracje typu mogą być dodawane do argumentów funkcji, wartości zwracanych oraz pól klas (od PHP 7.4.0).

typy argumentów

```
<?php  
function dodaj(int $a, int $b): int {  
    return $a + $b;  
}
```

typ zwracany

```
echo dodaj(4,5); //9  
?>
```

wywołanie funkcji

# Zmienne - rzutowanie

```
<?php  
$liczba = 0;  
echo var_dump($liczba); //int(0)
```

rzutowanie (cast) na typ boolean

```
$b = (boolean) $liczba; //bool(false)  
echo var_dump($b);
```

\$liczba nie zmienia typu

```
echo var_dump($liczba); //int(0)  
?>
```

dozwolone typy rzutowania

(int), (integer) - cast to int  
(bool), (boolean) - cast to bool  
(float), (double), (real) - cast to float  
(string) - cast to string  
(array) - cast to array  
(object) - cast to object  
(unset) - cast to NULL

# Zmienne – zmiana typu

```
<?php  
$liczba = 10.12;  
echo var_dump($liczba); //float(10.12)
```

\$liczba trwale zmienia typ

```
settype($liczba,'string'); //string(5) "10.12"  
echo var_dump($liczba);
```

\$liczba trwale zmienia typ

```
settype($liczba,'integer'); //int(10)  
echo var_dump($liczba);
```

```
?>
```

# String //typ łańcuchowy

ciąg znaków (napis), w którym każdy znak jest zapamiętany w jednym bajcie

Ciąg znaków można utworzyć używając:

- znaki apostrofu  
`echo 'to jest string';`
- znaki cudzysłowiu  
`echo "to jest string";`
- składni *heredoc*
- składni *nowdoc*

# składnia *heredoc*

```
<!--doctype html-->
```

```
<html>
```

```
<head>
```

```
    <title>heredoc</title>
```

```
    <meta charset="utf-8">
```

```
</head>
```

```
<body> obowiązkowe
```

identyfikator  
(zasady jak nazwa zmiennej)

```
<?php  
$tekst = <<<ETYKIETA
```

```
<p>zakupy:</p>
```

```
<ul>
```

```
    <li>jabłka</li>
```

```
    <li>gruszki</li>
```

```
</ul>
```

```
ETYKIETA; ← tylko identyfikator  
            oraz średnik  
            (przed ETYKIETA  
            nie może być spacji)
```

```
</body>
```

```
</html>
```



```
1 <!--doctype html-->
```

```
2 <html>
```

```
3 <head>
```

```
    <title>heredoc</title>
```

```
    <meta charset="utf-8">
```

```
6 </head>
```

```
7 <body>
```

```
8 <p>zakupy:</p>
```

```
9 <ul>
```

```
    <li>jabłka</li>
```

```
    <li>gruszki</li>
```

```
12 </ul></body>
```

```
13 </html>
```

```
14
```

Tak samo  
sformatowane



zakupy:



- jabłka
- gruszki

# składnia *heredoc*

```
<!doctype html>
<html>
<head>
    <title>heredoc</title>
    <meta charset="utf-8">
</head>
<body>
<?php
$a = 1;
$b = 2;
$tekst = <<<TEKST
pierwsza linijka $a
druga linijka $b
TEKST;
echo $tekst;
?>
</body>
</html>
```

Tak samo  
sformatowane



```
<!doctype html>
<html>
<head>
    <title>heredoc</title>
    <meta charset="utf-8">
</head>
<body>
    pierwsza linijka 1
    druga linijka 2</body>
</html>
```

tutaj mamy znak nowej linii '\n' a nie <br>



pierwsza linijka 1 druga linijka 2

przeglądarka wyświetla w jednej linii

# składnia *heredoc*

```
<!doctype html>
<html>
<head>
    <title>heredoc</title>
    <meta charset="utf-8">
</head>
<body>
<?php
$a = 1;
$b = 2;
$tekst = <<<TEKST
pierwsza linijka $a
druga linijka $b
TEKST;
echo nl2br($tekst);
?>
</body>
</html>
```



```
<!doctype html>
<html>
<head>
    <title>heredoc</title>
    <meta charset="utf-8">
</head>
<body>
    pierwsza linijka 1<br />
    druga linijka 2</body>
</html>
```



pierwsza linijka 1  
druga linijka 2

'\n' => <br>

# składnia *nowdoc* (różnice względem heredoc)

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <title>heredoc</title>
```

```
    <meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
<?php obowiązkowe identyfikator  
$a = 1; w apostrofach
```

```
$b = 2;  
$tekst = <<<'TEKST'  
pierwsza linijka $a  
druga linijka $b
```

```
TEKST;  
echo nl2br($tekst);
```

```
?>
```

```
</body>
```

```
</html>
```

Tak samo  
sformatowane

tylko identyfikator  
oraz średnik  
(przed TEKST  
nie może być spacji)

'\n' => <br>

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <title>heredoc</title>
```

```
    <meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
pierwsza linijka $a<br />  
druga linijka $b</body>
```

```
</html>
```



pierwsza linijka \$a  
druga linijka \$b

nie printuje wartości zmiennych

# Integer //liczba całkowita (ujemna lub dodatnia)

```
<?php
$liczba1 = 1234; // system dziesiętny - podstawa 10 (decimal, base 10)
$liczba2 = 1_234_567; // system dziesiętny (od PHP 7.4.0)
$liczba3 = 0123; // system ósemkowy - podstawa 8 (octal, base 8)
$liczba4 = 0x1A; // system szesnastkowy - podstawa 16 (hexadecimal, base 16) )
$liczba5 = 0b11111111; // system binarny - podstawa 2 (binary, base 2)
?>
```

echo PHP\_INT\_MAX

9223372036854775807

echo PHP\_INT\_MIN

-9223372036854775808

wartości zależą od sprzętu  
i systemu

<https://www.php.net/manual/en/language.types.integer.php>

## Integer - przekroczenie zakresu liczby

```
<?php
$liczbaMax = 9223372036854775807;
var_dump($liczbaMax); // int(9223372036854775807)

$liczbaMaxPlusJeden = 9223372036854775808;
var_dump($liczbaMaxPlusJeden); // float(9.2233720368548E+18)
?>
```

# Float //liczba zmiennoprzecinkowa (ujemna lub dodatnia)

```
<?php
$liczba1 = 1.234;
$liczba2 = 1.2e3;      // notacja naukowa 1.2 * 10^3
$liczba3 = 7E-10;      // notacja naukowa 7 * 10^-10
$liczba4 = 1_234.567; // od PHP 7.4.0
?>
```

echo PHP\_FLOAT\_MAX

1.7976931348623E+308

echo PHP\_FLOAT\_MIN

2.2250738585072E-308

wartości zależą od sprzętu  
i systemu

<https://www.php.net/manual/en/language.types.float.php>

# Konwersja dec to bin

W przypadku Integera każda liczba dec może być dokładnie reprezentowana w bin.  
Problem jest w przypadku niektórych ułamków:

$$0.1_{10} \Rightarrow 000(1100)_2$$

liczba okresowa, nieskończona

```
<?php
$liczba = floor((0.1 + 0.7) * 10);
echo $liczba;
?>
```

reprezentacja liczby dziesiętnej w komputerze  
(w tym przypadku musi zostać zaokrąglona)

// 7

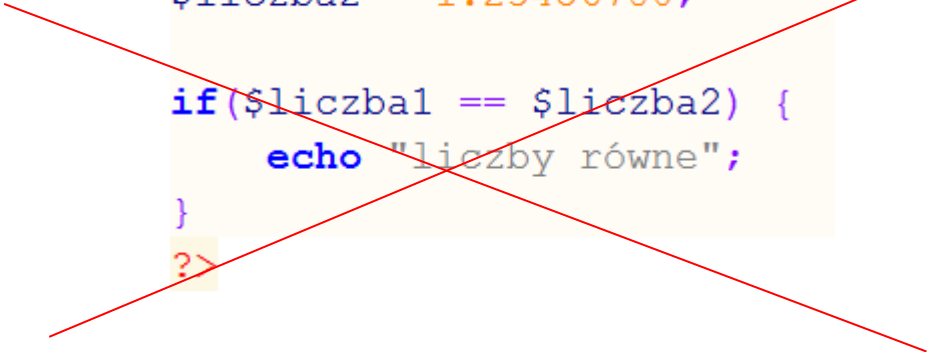


powielamy błąd

[https://books.google.pl/books?id=fKE68pdJCD4C&pg=PA14&lpg=PA14&dq=fractions+dec+to+bin+problems&source=bl&ots=nUXrx6yo6-&sig=ACfU3U2UfyAvmM4QdXYpF5DNpjECmUK0Xw&hl=pl&sa=X&ved=2ahUKEwi\\_hbLZyIP0Ah\\_VHAXAIHVW7AJwQ6AF6BAgeEAM#v=onepage&q=fractions%20dec%20to%20bin%20problem&s&f=false](https://books.google.pl/books?id=fKE68pdJCD4C&pg=PA14&lpg=PA14&dq=fractions+dec+to+bin+problems&source=bl&ots=nUXrx6yo6-&sig=ACfU3U2UfyAvmM4QdXYpF5DNpjECmUK0Xw&hl=pl&sa=X&ved=2ahUKEwi_hbLZyIP0Ah_VHAXAIHVW7AJwQ6AF6BAgeEAM#v=onepage&q=fractions%20dec%20to%20bin%20problem&s&f=false)

# Float - porównanie

liczby typu Float



```
<?php
$liczba1 = 1.23456789;
$liczba2 = 1.23456780;

if($liczba1 == $liczba2) {
    echo "liczby równe";
}
?>
```

<https://www.php.net/manual/en/language.types.float.php>

# Float - porównanie

```
<?php
$liczba1 = 1.23456789;
$liczba2 = 1.23456780;
$epsilon = 0.00001;

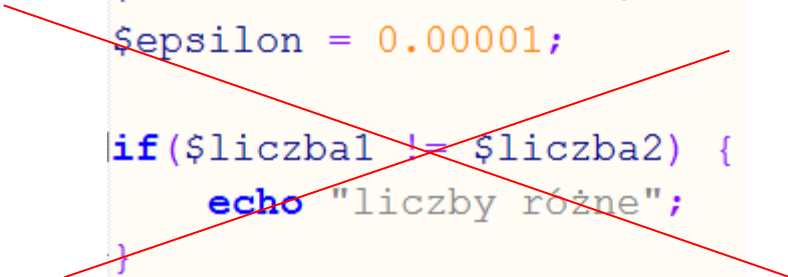
if(abs($liczba1-$liczba2) <=$epsilon) {
    echo "liczby prawie równe";           // liczby prawie równe
}
?>
```

# Float - porównanie

```
<?php
$liczba1 = 1.2345;
$liczba2 = 1.23456780;
$epsilon = 0.00001;

if($liczba1 != $liczba2) {
    echo "liczby różne";
}

?>
```



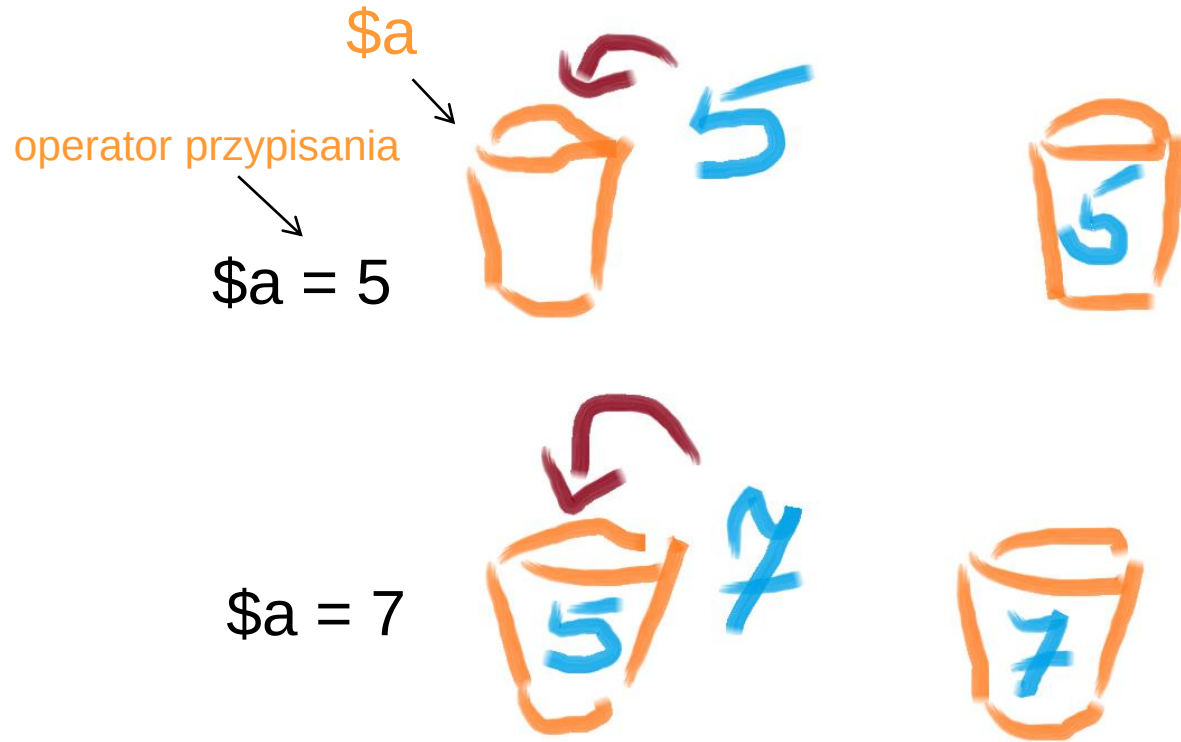
# Float - porównanie

```
<?php
$liczba1 = 1.2345;
$liczba2 = 1.23456780;
$epsilon = 0.00001;

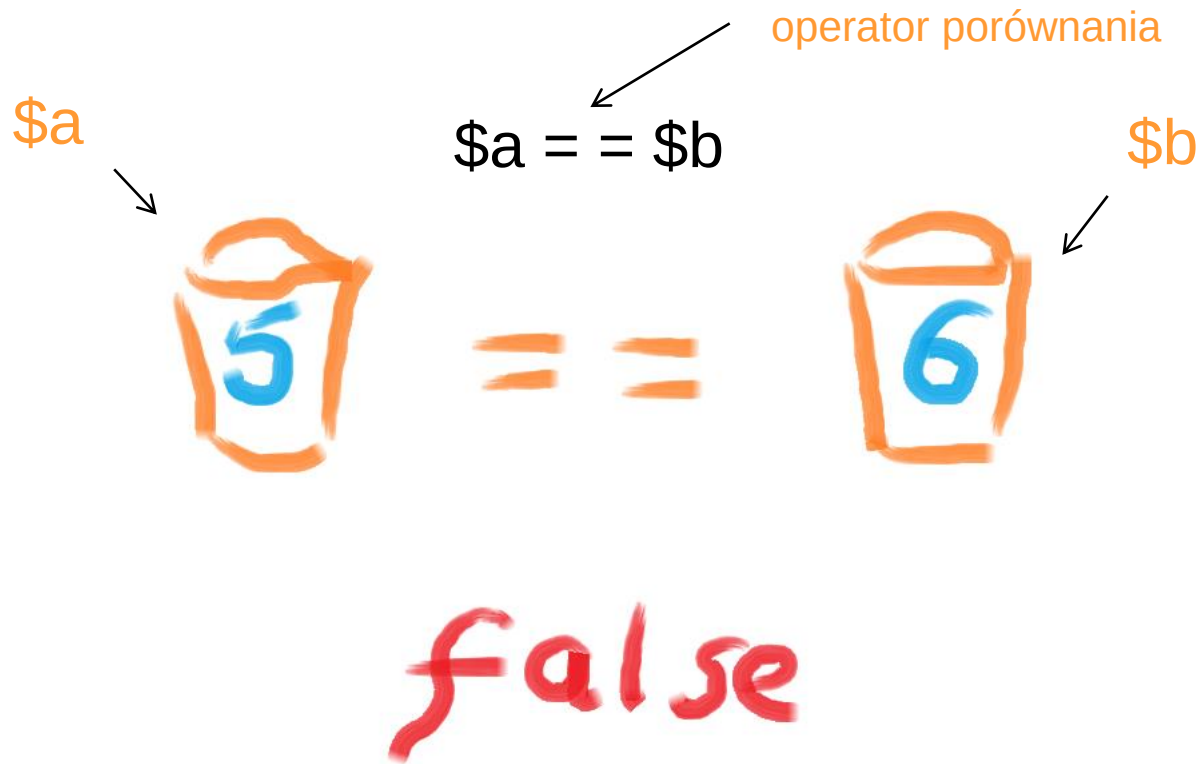
if(abs($liczba1 - $liczba2) > $epsilon) {
    echo "liczby różne";
}

?>
```

# Operator przypisania



# Operator porównania



# Zmienne

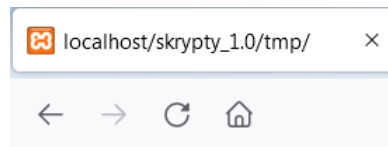
definicja i inicjalizacja zmiennej a

`$a=10;`

operator przypisania

example

```
<?php  
$a=10;  
echo $a;  
?>
```



10

output of example

# Zakres globalny i lokalny

zmienna globalna – zasięg cały skrypt  
(z wyjątkiem wnętrza funkcji)

zmienna globalna  
nie jest widoczna  
we wnętrzu funkcji

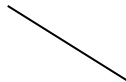
```
<?php
// zakres globalny (widoczność poza funkcją)
$liczba = 10;

// definicja funkcji wypisz
function wypisz()
{
    // referencja do zmiennej lokalnej (której nie ma)
    echo $liczba;
}

wypisz(); // nic nie wypisze
?>
```

# Zakres globalny i lokalny

zmienna globalna  
jest widoczna  
we wnętrzu funkcji



```
<?php
// zakres globalny (widoczność poza funkcją)
$liczba = 10;

// definicja funkcji wypisz
function wypisz()
{
    // deklaracja $liczba jako zmiennej globalnej
    global $liczba;

    echo $liczba;
}

wypisz(); // wypisze 10
?>
```

# Zakres globalny i lokalny

zmienna lokalna

zmienna globalna

```
<?php
// zmienna globalna (widoczność poza funkcją)
$liczba = 10;
function wypisz(){
    // zmienna lokalna - zasięg wewnątrz funkcji
    $liczba = 20;
    echo $liczba;
}
// wypisuje zmienna lokalną
wypisz(); //20
// zmienna globalna
echo $liczba; //10
?>
```

# Zmienna lokalna

```
<?php
function wypisz(){
    // zmienna lokalna (nie statyczna)
    // jej wartość nie zostaje zachowana
    // pomiędzy wywołaniami funkcji
    $liczba = 1;
    echo $liczba."<br>";
    // inkrementacja zmiennej lokalnej
    $liczba++;
}
// wypisuje zmienną lokalną
wypisz(); //1
wypisz(); //1
wypisz(); //1
?>
```

zmienna lokalna  
(nie statyczna)  
nie zachowuje swojej wartości  
pomiędzy wywołaniami  
funkcji

kolejne wywołania funkcji  
wypisują 1

# Zmienna statyczna

```
<?php
function wypisz(){
    // zmienna statyczna - jej wartość zostaje zachowana
    // pomiędzy wywołaniami funkcji
    static $liczba = 1;
    echo $liczba."<br>";
    // inkrementacja zmiennej statycznej
    $liczba++;
}
// wypisuje zmienną statyczną
wypisz(); //1
wypisz(); //2
wypisz(); //3
?>
```

zmienna statyczna  
zachowuje swoją wartość  
pomiędzy wywołaniami  
funkcji

ten wiersz zostaje pominięty  
przy kolejnych wywołaniach  
funkcji

kolejne wywołania funkcji  
wypisują kolejne liczby

# zmienne predefiniowane

`$_GET` – tablica zawierająca zmienne dostarczane do skryptu za pomocą metody GET protokołu HTTP

`$_POST` – tablica zawierająca zmienne dostarczane do skryptu za pomocą metody POST protokołu HTTP

`$_COOKIE` – tablica zawierająca zmienne dostarczane do skryptu za pomocą ciasteczek HTTP

`$_SESSION` – tablica zawierająca zmienne aktualnie zarejestrowane jako sesyjne

`$_SERVER` – tablica zawierająca zmienne dostarczane do skryptu z serwera HTTP

`$_FILES` – tablica zawierająca zmienne z informacjami o plikach wysłanych do serwera metodą POST

`$_ENV` – tablica zawierająca informacje o zmiennych środowiskowych serwera

`$_REQUEST` – tablica zawierająca dane z `$_GET`, `$_POST` oraz `$_COOKIE`

`$_GLOBALS` – referencje (odniesienia) do zmiennych globalnych (zdefiniowanych przez użytkownika w zakresie globalnym)

# Zmienne zmienne

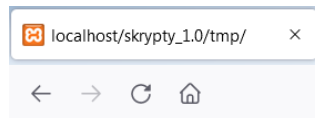
`$a="liczba";`

`$$a`  
`${$a}`



`$liczba`

```
<?php
$a='liczba';
$$a=5;
echo $liczba;
?>
```

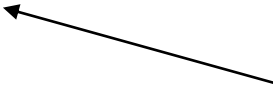


5

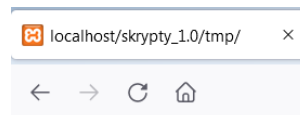
# Zmienne zmienne

```
for ($i = 0; $i < 2; $i++) {  
    ${'a'.$i} = $i;  
}
```

\$a0=0  
\$a1=1



```
<?php  
for ($i = 0; $i < 2; $i++) {  
    ${'a'.$i} = $i;  
}  
echo $a0." ".$a1;  
?>
```



0 1

# Stałe Constants

identyfikator, którego wartość jest stała

definicja stałej KOLOR

```
<?php  
define("KOLOR", "czerwony");  
  
echo constant("KOLOR"); //czerwony  
  
echo "róża ma ".KOLOR." kolor"; //róża ma czerwony kolor  
?>
```

funkcja constant zwraca wartość stałej

# Stałe predefiniowane *Predefined Constants*

```
<?php
echo TRUE;           //1
echo FALSE;          //0
echo PHP_VERSION;    //8.0.10
echo PHP_OS;          //WINNT
echo __FILE__;        //C:\xampp\htdocs\html\...\skrypt.php
echo __LINE__;        //15
echo PHP_INT_MAX;     //159223372036854775807
echo PHP_INT_SIZE;    //8
?>
```

nazwa  
przetwarzanego  
skryptu

numer  
przetwarzanej  
linii

wielkość integera  
w B

# Zmienne zmienne

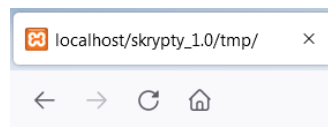
```
$tab[0]=„liczba0”;  
$tab[1]=„liczba1”;
```

```
${$tab[0]}=5;  
${$tab[1]}=6;
```

**\$liczba0=5**

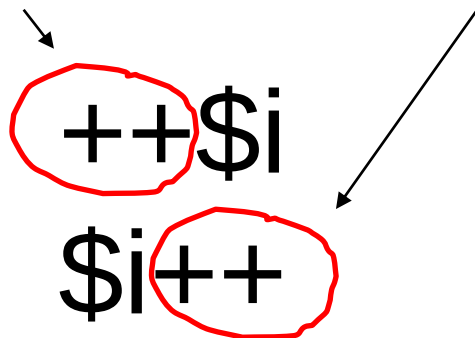
**\$liczba1=6**

```
<?php  
$tab[0]="liczba0";  
$tab[1]="liczba1";  
${$tab[0]}=5;  
${$tab[1]}=6;  
echo $liczba0." ".$liczba1;  
?>
```



5 6

# Operatory preinkrementacji i postinkrementacji

  
++\$i  
\$i++

```
$i=1;  
$a=1;  
$a=++$i;  
echo $a; //2  
echo „<br>”;  
echo $i; //2
```

```
$i=1;  
$a=1;  
$a=$i++;  
echo $a; //1  
echo „<br>”;  
echo $i; //2
```

# Operatory predekrementacji i postdekrementacji

--\$i

\$i--

```
$i=1;  
$a=1;  
$a=--$i;  
echo $a; //0  
echo „<br>”;  
echo $i; //0
```

```
$i=1;  
$a=1;  
$a=$i--;  
echo $a; //1  
echo „<br>”;  
echo $i; //0
```

# Operatory inkrementacji oraz dekrementacji

przykłady dla  $x=4$ ;

|       |                    |          |             |
|-------|--------------------|----------|-------------|
| $++x$ | pre-inkrementacja  | $y=++x;$ | $y=5; x=5;$ |
| $--x$ | pre-dekrementacja  | $y=--x;$ | $y=3; x=3;$ |
| $x++$ | post-inkrementacja | $y=x++;$ | $y=4; x=5;$ |
| $x--$ | post-dekrementacja | $y=x--;$ | $y=4; x=3;$ |

# Operatory arytmetyczne

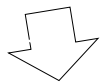
|              |             |                                                                                                                                        |                       |
|--------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| $+\$a$       | tożsamość   | zwraca $\$a$                                                                                                                           | <i>Identity</i>       |
| $-\$a$       | nagacja     | zwraca liczbę przeciwną                                                                                                                | <i>Negation</i>       |
| $\$a + \$b$  | dodawanie   | suma $\$a$ oraz $\$b$                                                                                                                  | <i>Addition</i>       |
| $\$a - \$b$  | odejmowanie | różnica $\$a$ oraz $\$b$                                                                                                               | <i>Subtraction</i>    |
| $\$a * \$b$  | mnożenie    | iloczyn $\$a$ oraz $\$b$                                                                                                               | <i>Multiplication</i> |
| $\$a / \$b$  | dzielenie   | iloraz $\$a$ oraz $\$b$ (jeśli $\$a$ oraz $\$b$ są typu integer wynik jest typu integer, w przeciwnym razie wynikiem jest typ float)   | <i>Division</i>       |
| $\$a \% \$b$ | modulo      | reszta z dzielenia $\$a$ przez $\$b$ , operandy są niejawnie konwertowane do integera, zwraca integer o znaku zgodnym ze znakiem $\$a$ | <i>Modulo</i>         |
| $\$a ** \$b$ | potęgowanie | $\$a$ do potęgi $\$b$ ( $a^b$ )                                                                                                        | <i>Exponentiation</i> |

# Operatory bitowe

|                  |                             |                           |
|------------------|-----------------------------|---------------------------|
| $\$a \& \$b$     | iloczyn bitowy              | <i>And</i>                |
| $\$a   \$b$      | suma bitowa                 | <i>Or (inclusive or)</i>  |
| $\$a \wedge \$b$ | bitowa różnica symetryczna  | <i>Xor (exclusive or)</i> |
| $\sim \$a$       | negacja bitowa              | <i>Not</i>                |
| $\$a \gg \$b$    | przesunięcie bitowe w prawo | <i>Shift right</i>        |
| $\$a \ll \$b$    | przesunięcie bitowe w lewo  | <i>Shift left</i>         |

# Operator bitowy AND

```
<?php
$a = 6;
$b = 4;
$c = $a & $b;
echo $c; //?
```



system binarny

6 = 110

4 = 100



110  
& 100

-----

100



| x | y | x & y |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 0     |
| 1 | 0 | 0     |
| 1 | 1 | 1     |

Tablica prawdy dla AND



system dziesiętny

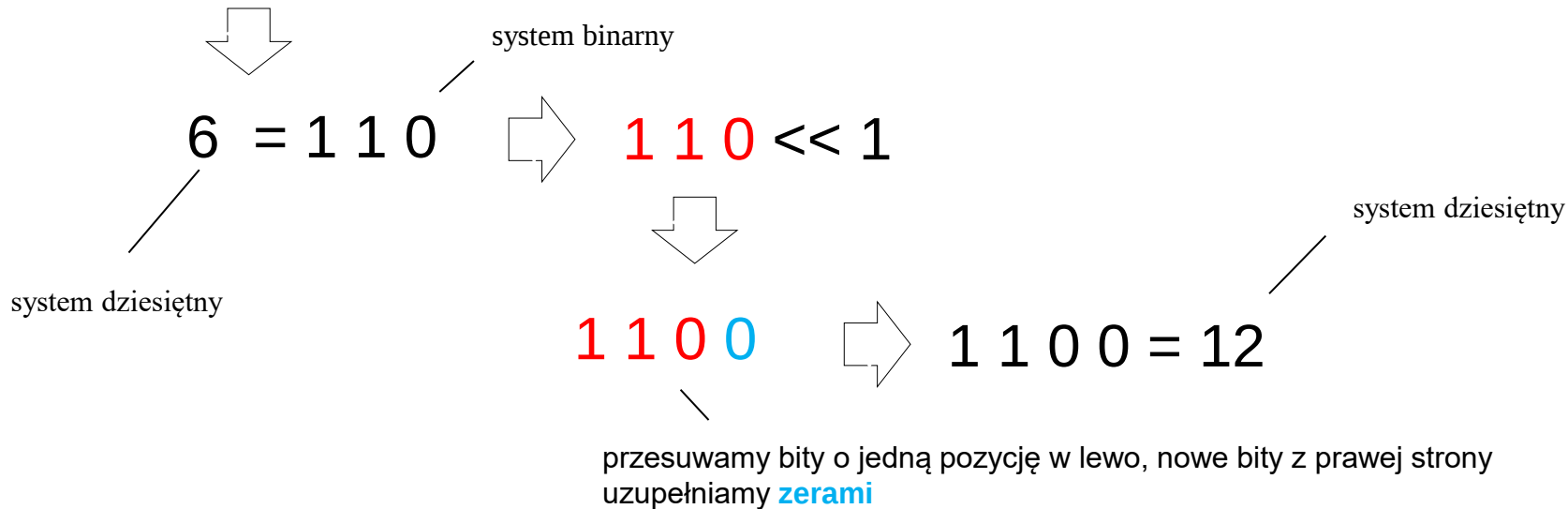


4

system dziesiętny

# Operator przesunięcia bitowego w lewo

```
<?php  
$a = 6;  
$c = $a << 1;  
echo $c;  //?  
?>
```



# Wyrażenie

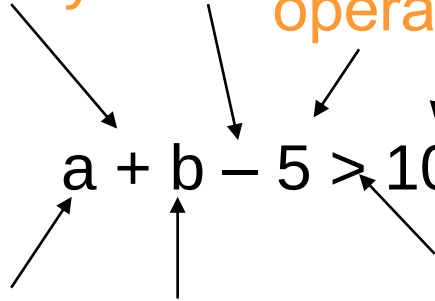
operatory arytmetyczne

operandy stałe

a + b - 5 > 10

operandy zmienne

operator relacji



# Operatory relacji

|                               |                   |                                                                         |                                 |
|-------------------------------|-------------------|-------------------------------------------------------------------------|---------------------------------|
| <code>\$a == \$b</code>       | równy             | true jeśli \$a jest równe \$b                                           | <i>Equal to</i>                 |
| <code>\$a === \$b</code>      | identyczny        | true jeśli \$a jest równe \$b oraz \$a i \$b są tego samego typu        | <i>Identical</i>                |
| <code>\$a != \$b</code>       | różny             | true jeśli \$a jest różny od \$b                                        | <i>Not equal to</i>             |
| <code>\$a &lt;&gt; \$b</code> | różny             | true jeśli \$a jest różny od \$b                                        | <i>Not equal to</i>             |
| <code>\$a !== \$b</code>      | nie identyczny    | true jeśli \$a nie jest równe \$b lub \$a i \$b nie są tego samego typu | <i>Not identical</i>            |
| <code>\$a &gt; \$b</code>     | większy           | true jeśli \$a jest większy od \$b                                      | <i>Greater than</i>             |
| <code>\$a &lt; \$b</code>     | mniej             | true jeśli \$a jest mniej od \$b                                        | <i>Less than</i>                |
| <code>\$a &gt;= \$b</code>    | większy lub równy | true jeśli \$a jest większy lub równy \$b                               | <i>Greater than or equal to</i> |
| <code>\$a &lt;= \$b</code>    | mniej lub równy   | true jeśli \$a jest mniej lub równy \$b                                 | <i>Less than or equal to</i>    |

# Instrukcja warunkowa if

Wyrażenie może przyjąć jedną z dwóch wartości logicznych:  
true - wyrażenie jest prawdziwe  
false – wyrażenie jest fałszywe

W PHP przyjmuje się, że:  
0 – false  
liczba różna od zera – true



```
if (wyrażenie)  
    instrukcja1;
```

Przykłady:

(a==2)

true dla zmiennej a równej 2

false dla zmiennej a równej 3

(a)

true dla zmiennej a różnej od 0 (np.: 1,2,4, 10 itp.)

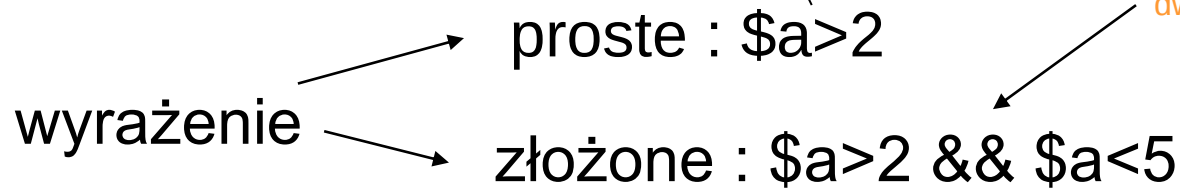
false dla zmiennej a równej 0

(a+2\*b>c)

true jeśli wartość wyrażenia a+2\*b jest większa od c

# Instrukcja warunkowa if

w wyrażeniach prostych możemy zastosować operatory relacji



koniunkcja (iloczyn logiczny)  
dwóch warunków

w wyrażeniach złożonych możemy zastosować operatory logiczne  
do połączenia kilku warunków

# Operatory logiczne *Logical operators*

|                                 |                     |                                                                           |            |
|---------------------------------|---------------------|---------------------------------------------------------------------------|------------|
| <code>\$a &amp;&amp; \$b</code> | iloczyn logiczny    | true, jeśli obydwie zmienne \$a and \$b mają wartość true.                | <i>And</i> |
| <code>\$a and \$b</code>        | iloczyn logiczny    | true, jeśli obydwie zmienne \$a and \$b mają wartość true.                | <i>And</i> |
| <code>\$a    \$b</code>         | suma logiczna       | true, jeśli któraś ze zmiennych \$a lub \$b ma wartość true               | <i>Or</i>  |
| <code>\$a or \$b</code>         | suma logiczna       | true, jeśli któraś ze zmiennych \$a lub \$b ma wartość true               | <i>Or</i>  |
| <code>!\$a</code>               | negacja logiczna    | true, jeśli \$a ma wartość false                                          | <i>Not</i> |
| <code>\$a xor \$b</code>        | różnica symetryczna | true, jeśli \$a oraz \$b mają różną wartość (np. \$a=true oraz \$b=false) | <i>Xor</i> |

# Instrukcja warunkowa if

if (wyrażenie)

→ instrukcja1;

pojedyncza instrukcja  
nie musi mieć nawiasów  
klamrowych

if (wyrażenie)

instrukcja1;

else

instrukcja2;

if (wyrażenie) {

instrukcja1;

instrukcja2;

instrukcja3;

}



blok instrukcji  
(musi być ujęty  
w nawiasy klamrowe)

if (wyrażenie1)

instrukcja1;

else if (wyrażenie2)

instrukcja2;

else if (wyrażenie3)

instrukcja3;



wybór wielowariantowy

if (wyrażenie1)

instrukcja1;

else if (wyrażenie2)

instrukcja2;

else if (wyrażenie3)

instrukcja3;

else

instrukcja4;

# Operator warunkowy

## Conditional operator

warunek ? wartość1 : wartość2;



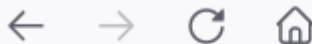
if (warunek)

wartość1;

e|se

wartość2;

```
<?php
$a = 23;
$jestLubNieJest = ($a % 2 == 0) ? "jest" : "nie jest";
echo "liczba $a $jestLubNieJest parzysta".<br>;
?>
```



liczba 23 nie jest parzysta

# Instrukcja switch

```
switch(wyrażenie)
{
    case wartość1:
        instrukcja1;
        break;
    case wartość2:
        instrukcja2;
        break;
    default:
        instrukcja3;
}
```

wyjście z instrukcji switch

# Priorytet operatorów

| Łączność        | Operatory                                                    | Dodatkowe informacje                                                                     |
|-----------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------|
| (n/a)           | clone new                                                    | clone and new                                                                            |
| right           | **                                                           | <a href="#">arithmetic</a>                                                               |
| (n/a)           | + - ++ -- ~ (int) (float) (string) (array) (object) (bool) @ | arithmetic (unary + and -), increment/decrement, bitwise, type casting and error control |
| left            | instanceof                                                   | <a href="#">type</a>                                                                     |
| (n/a)           | !                                                            | <a href="#">logical</a>                                                                  |
| left            | * / %                                                        | <a href="#">arithmetic</a>                                                               |
| left            | + - .                                                        | arithmetic (binary + and -), array and string (. prior to PHP 8.0.0)                     |
| left            | << >>                                                        | <a href="#">bitwise</a>                                                                  |
| left            | .                                                            | <a href="#">string (as of PHP 8.0.0)</a>                                                 |
| non-associative | < <= > >=                                                    | <a href="#">comparison</a>                                                               |
| non-associative | == != === !== <= < > >=                                      | <a href="#">comparison</a>                                                               |
| left            | &                                                            | bitwise and references                                                                   |
| left            | ^                                                            | <a href="#">bitwise</a>                                                                  |
| left            |                                                              | <a href="#">bitwise</a>                                                                  |
| left            | &&                                                           | <a href="#">logical</a>                                                                  |
| left            |                                                              | <a href="#">logical</a>                                                                  |
| right           | ??                                                           | <a href="#">null coalescing</a>                                                          |
| non-associative | ? :                                                          | <a href="#">ternary (left-associative prior to PHP 8.0.0)</a>                            |
| right           | = += -= *= **= /= .= %= &=  = ^= <<= >>= ??=                 | <a href="#">assignment</a>                                                               |
| (n/a)           | yield from                                                   | <a href="#">yield from</a>                                                               |
| (n/a)           | yield                                                        | <a href="#">yield</a>                                                                    |
| (n/a)           | print                                                        | <a href="#">print</a>                                                                    |
| left            | and                                                          | <a href="#">logical</a>                                                                  |
| left            | xor                                                          | <a href="#">logical</a>                                                                  |
| left            | or                                                           | <a href="#">logical</a>                                                                  |

# Priorytet operatorów

```
<?php
$a = 2 * 4 + 5; // 13 mnożenie ma większy priorytet niż dodawanie
$b = 3 * 3 % 5; // 4 ten sam priorytet, łączność lewostronna (działanie wykonujemy od lewej do prawej)
$c = 10 - 6 + 4; // 8 ten sam priorytet, łączność lewostronna (działanie wykonujemy od lewej do prawej)
$d = 1;
$e = 2;
$d = $e += 3; // ten sam priorytet, łączność prawostronna, $d = ($e += 3) -> $d = 5, $e = 5
echo $a."<br>";
echo $b."<br>";
echo $c."<br>";
echo $d."<br>";
echo $e."<br>";
?>
```

# Priorytet operatorów

```
<?php
$a = 8 * 2 - 5;    //?
$b = 10 % 3 - 7;
$c = 10 - 6 - 2 + 4;
$d = 1;
$e = 2;
$d = $e *= 3;
$f = 1;
$g = ++$f - 4;
$h = 1;
$i = $h-- - 3;
?>
```

# Pętla for

Przed pierwszym uruchomieniem pętli wykonywana jest ta instrukcja

Gdy jest tylko jedna instrukcja nawiasów może nie być

Warunek pętli – wyrażenie sprawdzane przed każdorazowym obiegiem pętli. Jeśli jest prawdziwe wykonywana jest treść pętli.

Instrukcja wykonywana każdorazowo na zakończenie obiegu pętli

```
for ($i=0; $i<10; $i++)  
{  
    echo „Licznik pętli: ”;  
    echo $i;  
    echo „<br>”  
}
```

Treść pętli (instrukcje ujęte w nawiasach klamrowych)

# Pętla for

Deklaracja i zainicjowanie licznika pętli

Warunek

Krok pętli

```
for ($i=0; $i<10; $i++)  
    echo $i;
```

Ciało pętli

Gdy jest tylko  
jedna instrukcja  
nawiasów klamrowych  
może nie być

# Pętla for

Kolejność wykonywania instrukcji

```
for ($i1=0; $i2<10; $i4++)  
{  
    echo $i; 3  
}
```

pierwszy obieg pętli 1234

drugi obieg pętli 234

trzeci obieg pętli 234

kolejne obiegi pętli 234

# Pętla while

dopóki jest spełniony

warunek

Gdy jest tylko  
jedna instrukcja  
nawiasów  
może nie być

while (wyrażenie)

{

instrukcja;

}

wykonuj instrukcje

wyrażenie może przyjmować wartość logiczną:

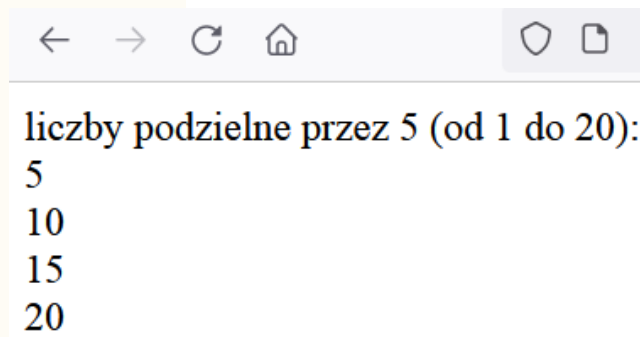
- prawda
- fałsz

# Pętla do while



# Instrukcja break

```
<?php
echo "liczby podzielne przez 5 (od 1 do 20):"."<br>";
// licznik petli
$i = 1;
// petla nieskończona
for(;;){
    // warunek podzielności przez 5
    if ($i % 5 == 0)
        echo $i."<br>";
    // inkrementacja licznika pętli
    $i++;
    // warunek wyjścia z pętli
    if ($i > 20)
        break;
}
?>
```

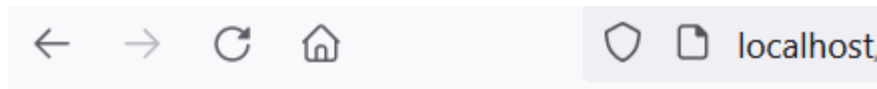
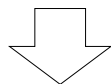


wyjście z pętli

# Instrukcja continue

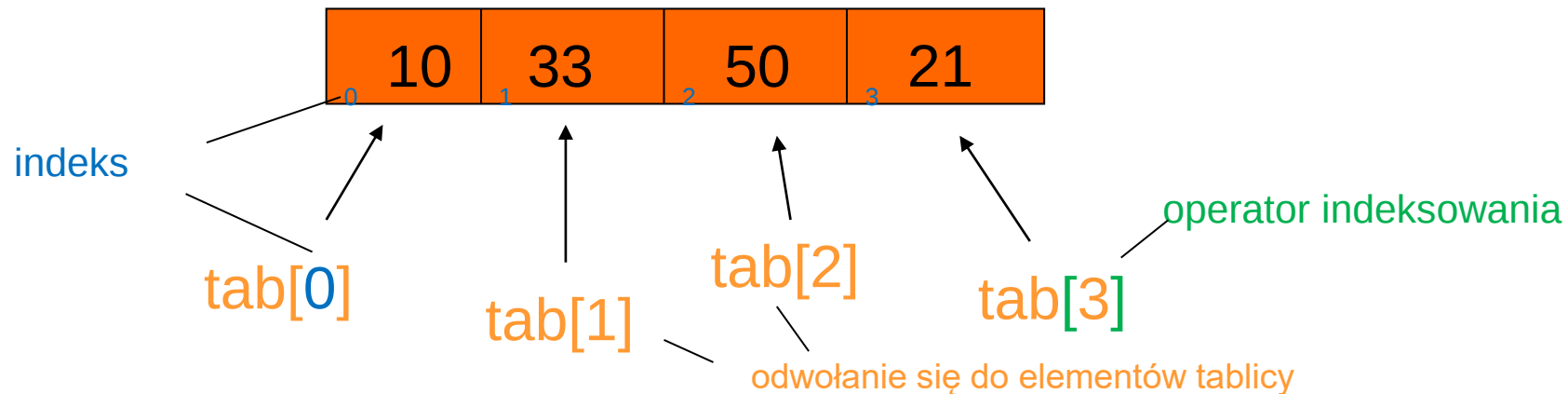
```
<?php
echo "liczby niepodzielne przez 5 (od 1 do 20):". "<br>";
// petla od 1 do 20
for($i = 1; $i <= 20; $i++){
    // pomijamy liczby podzielne przez 5
    if ($i % 5 == 0)
        continue;
    echo $i." ";
}
?>
```

wraca na początek pętli



liczby niepodzielne przez 5 (od 1 do 20):  
1 2 3 4 6 7 8 9 11 12 13 14 16 17 18 19

# Tablica jednowymiarowa



# Tablica jednowymiarowa

```
<?php
$tab[0] = 10;
$tab[1] = 33;
$tab[2] = 50;
$tab[3] = 21;

$tab1[] = 10;
$tab1[] = 33;
$tab1[] = 50;
$tab1[] = 21;

$tab2 = array(10,33,50,21);

$tab3 = [10,33,50,21];

echo "<h1>$tab[2]</h1>"; //50
echo "<h1>$tab1[2]</h1>"; //50
echo "<h1>$tab2[2]</h1>"; //50
echo "<h1>$tab3[2]</h1>"; //50

?>
```

różne sposoby inicjalizacji tablicy



# Tablice

inicjalizowanie tablicy \$tab[]

```
for ($i=0; $i<4; $i++)  
    $tab[$i]=$i;
```

```
$tab[0] = 0;  
$tab[1] = 1;  
$tab[2] = 2;  
$tab[3] = 3;
```

wypisanie zawartości tablicy

```
for ($i=0; $i<4; $i++)  
    echo $tab[$i]."<br>";
```

```
0  
1  
2  
3
```

# Tablice asjacyjne

identyfikatory znakowe (klucze)



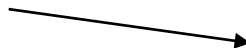
```
$tab["imie"] = "Ala";  
$tab["nazwisko"] = "Kowalska";  
$tab["klasa"] = "Ia";
```

przechowują pary klucz-wartość

```
$tab=array("imie" => "Ala",  
"nazwisko"=>"Kowalska",  
"klasa"=>"Ia");
```

alternatywnie

```
echo "moje dane:<br>";  
echo $tab["imie"]. "<br>";  
echo $tab["nazwisko"] . "<br>";  
echo $tab[„klasa"] . "<br>";
```



moje dane:  
Ala  
Kowalska  
Ia

# Funkcja array()

tworzy tablicę

automatyczna inkrementacja klucza

```
$m = array(1=>'styczeń', 'luty', 'marzec');  
print_r($m);
```

wyświetla informacje o zmiennej

Array

```
(  
[1] => Styczeń  
[2] => Luty  
[3] => Marzec  
)
```

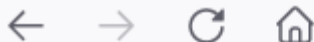
<https://www.php.net/manual/en/function.array.php>

[funkcje tablic](#)

# Tablica dwuwymiarowa

```
<?php
$tab=array(
    "owoce"    => array("gruszki" => "zielone", "jabłka" => "czerwone"),
    "warzywa" => array("pomidory" => "czerwone", "marchewki"=>"pomarańczowe"),
    "ilość"    => array("gruszki" => 20, "jabłka" =>3, "pomidory"=>10, "marchewki"=>5)
);

echo "kolory owoców i warzyw:<br>";
echo "gruszki: ".$tab["owoce"]["gruszki"].", ilość: ".$tab["ilość"]["gruszki"]."<br>";
echo "marchewki: ".$tab["warzywa"]["marchewki"].", ilość: ".$tab["ilość"]["marchewki"]."<br>";
?>
```



kolory owoców i warzyw:

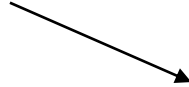
gruszki: zielone, ilość: 20

marchewki: pomarańczowe, ilość: 5

# Perl foreach

```
$tab=array("1", "2", "3");
```

```
foreach ($tab as $a){  
    echo "$a<br>";  
}
```



1  
2  
3

# Pętla foreach

```
<?php
// definiujemy tablicę
$tab[0] = 0;
$tab[1] = 1;
$tab[2] = 2;
$tab[3] = 3;

// ?
foreach($tab as &$liczba)
    $liczba = $liczba * (-1);
unset($liczba);

// wyświetlamy tablicę
foreach($tab as $liczba)
    echo $liczba."<br>";
?>
```

co tu się dzieje?

# Pętla foreach

```
<?php
// definiujemy tablicę
$tab[0] = 0;
$tab[1] = 1;
$tab[2] = 2;
$tab[3] = 3;

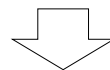
// ?
foreach($tab as &$liczba)
    $liczba = $liczba * (-1);
unset($liczba);

// wyświetlamy tablicę
foreach($tab as $liczba)
    echo $liczba."<br>";
?>
```

usuniecie zmiennej \$liczba

Przy każdej iteracji pętli wartość elementu tablicy jest przekazywana przez referencję do zmiennej \$liczba, która w ciele pętli jest mnożona przez liczbę -1.

W rezultacie następuje zamiana wszystkich elementów tablicy na liczby z przeciwnym znakiem



0  
-1  
-2  
-3

# Funkcja bez parametrów

słowo kluczowe

nazwa funkcji zgodna z zasadami nazewnictwa zmiennych

brak parametrów

ciało funkcji

```
<?php
function komunikat() {
    echo "Hello world!";
}

komunikat(); // Hello world!
?>
```

wywołanie funkcji

# Funkcja z jednym parametrem

jeden parametr przekazany przez wartość

```
<?php
function powitanie($imie) {
    echo "Hello $imie!";
}

powitanie("Marek"); // Hello Marek!
?>
```

ciąg tekstowy „Marek” jest kopiowany do \$imie (wewnątrz funkcji pracujemy na kopii argumentu funkcji)

argument funkcji

wywołanie funkcji z jednym argumentem

# Funkcja z argumentem domyślnym

```
<?php
function powitanie($imie = "Adam"){
    echo "Hello $imie!";
}
powitanie();           // Hello Adam!
powitanie("Ania");    // Hello Ania!
?>
```

argument domyślny

gdy nie podamy argumentu  
w wywołaniu funkcji  
wykorzystany zostanie  
argument domyślny

# Funkcja z dwoma parametrami

dwa parametry przekazywane przez wartość

```
<?php
function powitanie($imie, $nazwisko) {
    echo "Hello $imie $nazwisko!";
}

powitanie("Marek", "Kowalski"); // Hello Marek Kowalski!
?>
```

wywołanie funkcji z dwoma argumentami

# Funkcja - przekazywanie argumentu przez wartość

```
<?php
function inkrementuj($number){
    $number += 1; //inkrementacja liczby
    echo "liczba wewnatrz funkcji: $number<br>";
}
$liczba = 5;
echo "liczba przed wywołaniem funkcji: $liczba<br>";
inkrementuj($liczba);
echo "liczba po wywołaniu funkcji: $liczba<br>";
?>
```

wewnątrz funkcji działamy na kopii \$liczba  
(kopiujemy \$liczba do \$number)

argument przekazywany do funkcji przez wartość

liczba przed wywołaniem funkcji: 5  
liczba wewnątrz funkcji: 6  
liczba po wywołaniu funkcji: 5

argument przekazywany do funkcji przez wartość  
nie zmienił swojej wartości (inkrementowaliśmy  
wewnątrz funkcji jego kopię)

# Funkcja - przekazywanie argumentu przez referencję

argument przekazywany do funkcji przez referencję

wewnątrz funkcji działamy na oryginalnej zmiennej \$liczba  
(nie kopiujemy \$liczba do \$number)

```
<?php
function inkrementuj(&$number){
    $number += 1; //inkrementacja liczby
    echo "liczba wewnatrz funkcji: $number<br>";
}
$liczba = 5;
echo "liczba przed wywołaniem funkcji: $liczba<br>";
inkrementuj($liczba);
echo "liczba po wywołaniu funkcji: $liczba<br>";
?>
```

liczba przed wywołaniem funkcji: 5  
liczba wewnątrz funkcji: 6  
liczba po wywołaniu funkcji: 6

argument przekazywany do funkcji przez referencję  
zmienił swoją wartość (inkrementowaliśmy  
wewnątrz funkcji oryginał liczby)

# Funkcja z deklaracją typu parametrów

php jest słabo typowanym językiem,  
ale od php 7 można dodać deklarację typu  
w argumentach funkcji

```
<?php  
function dodaj(int $a, int $b) {  
    return $a + $b;  
}  
  
echo dodaj(5,3); // 8  
?>
```

wartość zwracana przez funkcję

# Funkcja z deklaracją typu parametrów

```
<?php
function dodaj(int $a, int $b) {
    return $a + $b;
}

echo dodaj(5, "3 dni"); // 8
?>
```

/

ponieważ zadeklarowaliśmy int \$b, ciąg tekstowy zostanie niejawnie skonwertowany na int(3)

# Funkcja z deklaracją typu parametrów

deklaracja musi być w pierwszej linijce w skrypcie

```
<?php  
declare(strict_types=1); // strict requirement  
function dodaj(int $a, int $b) {  
    return $a + $b;  
}  
  
echo dodaj(5, "3 dni");  
?>
```

Fatal error: Uncaught TypeError: dodaj():  
Argument #2 (\$b) must be of type int, string given,

ponieważ zadeklarowaliśmy `strict_types`, zostanie wyświetlony błąd  
(ponieważ ciąg tekstowy nie jest typu `int`)

# Funkcje matematyczne

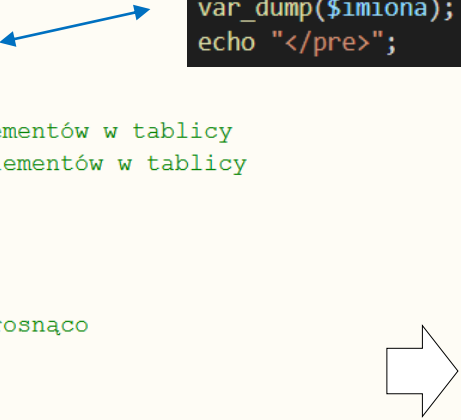
funkcje wbudowane

```
<?php
echo abs(-16);           // 16
echo floor(4.5);         // 4
echo ceil(4.5);          // 5
echo max(4, 2, 0, 8, 7); // 8
echo pi();               // 3.1415926535898
echo pow(2, 3);          // 8
?>
```

# Funkcje tablic

```
<?php
$imiona = array("Jan", "Ania", "Władysław", "Stefan", "Maria");
echo "tablica imiona:<br>";
foreach ($imiona as $key => $val) {
    echo "imiona[" . $key . "] = " . $val . "<br>";
}

$ilosc = count($imiona); // count() zwraca ilość elementów w tablicy
$ilosc1 = sizeof($imiona); // sizeof() zwraca ilość elementów w tablicy
echo "count():<br>";
echo "tablica zawiera $ilosc imion<br>";
echo "sizeof():<br>";
echo "tablica zawiera $ilosc1 imion<br>";
sort($imiona); // sort() sortuje tablicę rosnąco
echo "sort():<br>";
foreach ($imiona as $key => $val) {
    echo "imiona[" . $key . "] = " . $val . "<br>";
}
rsort($imiona);
echo "rsort():<br>"; // rsort() sortuje tablicę malejąco
foreach ($imiona as $key => $val) {
    echo "imiona[" . $key . "] = " . $val . "<br>";
}
echo "in_array():<br>"; // in_array() sprawdza, czy tablica zawiera dany element
if (in_array("Ania", $imiona)){
    echo "Ania jest w tablicy";
}
else{
    echo "Ani nie ma w tablicy";
}
?>
```



```
echo "<pre>";
var_dump($imiona);
echo "</pre>";
```

[funkcje tablic](#)

[funkcje sortujące tablic](#)



tablica imiona:

imiona[0] = Jan

imiona[1] = Ania

imiona[2] = Władysław

imiona[3] = Stefan

imiona[4] = Maria

count():

tablica zawiera 5 imion

sizeof():

tablica zawiera 5 imion

sort():

imiona[0] = Ania

imiona[1] = Jan

imiona[2] = Maria

imiona[3] = Stefan

imiona[4] = Władysław

rsort():

imiona[0] = Władysław

imiona[1] = Stefan

imiona[2] = Maria

imiona[3] = Jan

imiona[4] = Ania

in array():

Ania jest w tablicy

# Funkcje tablic asocjacyjnych

```
<?php
$wiek = array("Jan"=>7,"Ania"=>13,"Władysław"=>12,"Stefan"=>8,"Maria"=>3);
echo "tablica wiek (klucz => wartość):<br>";
foreach ($wiek as $key => $val) {
    echo "wiek[" . $key . "] = " . $val . "<br>";
}
asort($wiek);           // asort() sortuje wg. wartości rosnąco
echo "asort():<br>";
foreach ($wiek as $key => $val) {
    echo "wiek[" . $key . "] = " . $val . "<br>";
}
arsort($wiek);
echo "arsort():<br>";    // arsort() sortuje wg. wartości malejąco
foreach ($wiek as $key => $val) {
    echo "wiek[" . $key . "] = " . $val . "<br>";
}
ksort($wiek);           // ksort() sortuje wg. klucza rosnąco
echo "ksort():<br>";
foreach ($wiek as $key => $val) {
    echo "wiek[" . $key . "] = " . $val . "<br>";
}
krsort($wiek);
echo "krsort():<br>";    // krsort() sortuje wg. klucza malejąco
foreach ($wiek as $key => $val) {
    echo "wiek[" . $key . "] = " . $val . "<br>";
}
?>
```



tablica wiek (klucz => wartość):

wiek[Jan] = 7  
wiek[Ania] = 13  
wiek[Władysław] = 12  
wiek[Stefan] = 8  
wiek[Maria] = 3

**asort():**

wiek[Maria] = 3  
wiek[Jan] = 7  
wiek[Stefan] = 8  
wiek[Władysław] = 12  
wiek[Ania] = 13

**arsort():**

wiek[Ania] = 13  
wiek[Władysław] = 12  
wiek[Stefan] = 8  
wiek[Jan] = 7  
wiek[Maria] = 3

**ksort():**

wiek[Ania] = 13  
wiek[Jan] = 7  
wiek[Maria] = 3  
wiek[Stefan] = 8  
wiek[Władysław] = 12

**krsort():**

wiek[Władysław] = 12  
wiek[Stefan] = 8  
wiek[Maria] = 3  
wiek[Jan] = 7  
wiek[Ania] = 13

# Funkcje daty i czasu

```
<?php
$time=time(); //zwraca liczbę sekund od początku epoki Unixa
// (1 stycznia 1970 00:00:00 GMT)

echo "data:<br>";
echo($time . "<br>");
echo "data:<br>";
echo(date("d.m.Y",$time)); // date() formatuje datę i czas
echo "<br>godzina:<br>";
echo(date("G.i.s",$time));
echo("<br>getdate():<br>");
$date=getdate(); // getdate() zwraca tablicę asocjacyjną z bieżącą datą
print_r($date);
echo ("<br>rok:<br>");
echo $date["year"]; // można dostać się do poszczególnych składników daty
echo ("<br>dzień tygodnia<br>");
echo $date["weekday"];
// godzina, minuta, sekunda, miesiąc, dzień miesiąca, rok
$timestamp = mktime(12, 00, 01, 10, 5, 2021); // mktime() zamienia podany czas na zmienną czasu
echo ("<br>zadana data w formacie: godzina.minuta.sekunda dzień,miesiąc.rok<br>");
echo date("G.i.s d.m.Y", $timestamp); // wykorzystanie zmiennej czasu
?>
```

data:  
1639764964

data:  
17.12.2021

godzina:  
19.16.04

getdate():  
Array ( [seconds] => 4 [minutes] => 16 [hours] => 19 [mday] => 17 [wday] => 5 [mon] => 12 [year] => 2021 [yday] => 350 [weekday] => Friday [month] => December [0] => 1639764964 )

rok:  
2021

dzień tygodnia  
Friday

zadana data w formacie: godzina.minuta.sekunda dzień,miesiąc.rok  
12.00.01 05.10.2021



[funkcje daty i czasu](#)

[opis parametrów funkcji date\(\)](#)

# Funkcje operujące na ciągach tekstowych

```
<?php
echo chr(65) . "<br>"; // zwraca znak ASCII o zadanym kodzie
$str = "Komputery są ok";
print_r (explode(" ", $str)); // zamienia string na tablicę asocjacyjną
echo ("<br>4B4F4D5055544552:<br>");
echo hex2bin("4B4F4D5055544552"); // zamienia kod hex w znaki ASCII
echo nl2br("\nPierwsza linijka.\nDruga linijka.\n"); // zamienia "\n" na "<br>"
parse_str("a=50&b=3", $tab); // zamienia string w zmienne i umieszcza je w tablicy
print_r($tab);
echo "<br>formatowanie w kolumnie:<br>";
$str = "Pan Tadeusz jest ciekawą lekturą do poduszki";
echo wordwrap($str, 20, "<br>"); // wstawia znak nowego wiersza w stringu po 20 znakach
// (ale nie dzieli wyrazów)
echo "<br>długość stringu \"komputer jest ok\" wynosi:<br>";
echo strlen("komputer jest ok"); // zwraca długość ciągu tekstowego
echo "<br>zamiana na duże litery:<br>";
echo strtoupper("komputer jest ok"); // zamienia małe litery na duże
$str1 = "      spacje na początku i na końcu      ";
echo "<br>bez funkcji trim(): " . $str1;
echo "<br>z funkcja trim(): " . trim($str1); // usuwa białe znaki z obu stron stringa
// (spacje, tabulację poziomą i pionową ,
// znak nowej linii, znak powrotu karetki, NUL)
// ltrim() - usuwa białe znaki na początku stringu
// rtrim() - usuwa białe znaki z końca stringu
?>
```

przy zastosowaniu funkcji trim() różnicę widać w źródle strony

• bez funkcji trim():

spacje na początku i na końcu

<br>z funkcja trim(): spacje na początku i na końcu

A

Array ( [0] => Komputery [1] => są [2] => ok )  
4B4F4D5055544552:

KOMPUTER

Pierwsza linijka.

Druga linijka.

Array ( [a] => 50 [b] => 3 )

formatowanie w kolumnie:

Pan Tadeusz jest

ciekawą lekturą do

poduszki

długość stringu "komputer jest ok" wynosi:

16

zamiana na duże litery:

KOMPUTER JEST OK

bez funkcji trim(): spacje na początku i na końcu

z funkcja trim(): spacje na początku i na końcu

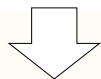
[PHP strings](#)

[PHP string reference](#)

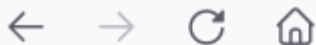
# Funkcje obsługi plików – include(), require()

```
<?php  
echo "Hello!";  
?>
```

plik hello.txt



```
<?php  
// dołączenie pliku do skryptu  
// w przypadku błędu, include zgłosi ostrzeżenie i skrypt nadal będzie działał  
include "hello.txt";  
  
// dołączenie pliku do skryptu  
// w przypadku błędu, require zgłosi błąd i zatrzyma skrypt  
require "hello.txt";  
?>
```



Hello!Hello!

# Funkcje obsługi plików - notatnik

<?php

```
// pole formularza
$tekst = $_POST["tekst"];
// nasz plik
$fileName = "plik.txt";
// tworzenie nowego pliku
//w: write - otwarcie pliku do zapisu, ustawia wskaźnik na początku pliku, gdy plik nie istnieje, tworzy go
//r: read - otwarcie pliku do odczytu, ustawia wskaźnik na początku pliku
//a: append - otwarcie pliku do zapisu (dopisania), ustawia wskaźnik na końcu pliku, gdy plik nie istnieje, tworzy go
$plik = fopen($fileName, "a") or die("Błąd otwarcia pliku do zapisu");
// dopisujemy przejście do nowej linii na końcu tekstu, żeby to co wpisujemy było w nowej linii
$tekst .= "\n";
// zapisanie tekstu w pliku
fwrite($plik, $tekst);
// zamknięcie pliku
fclose($plik);
// otwarcie pliku do odczytu
$plik = fopen($fileName, "r") or die("Błąd otwarcia pliku do odczytu");
// pobieramy zawartość pliku
$zawartosc = fread($plik, filesize($fileName));
// komunikaty
echo "do zawartości pliku dopisano: $tekst<br>";
echo "oto zawartość pliku:<br>";
// wyświetlamy zawartość (znak nowej linii w pliku zamieniamy na <br>)
echo nl2br($zawartosc);
fclose($plik);
```

action.php

Zadanie:  
dokończyć notatnik

?>

# Funkcje obsługi plików

## Example

```
<?php
echo readfile("webdictionary.txt");
?>
```

Run example »

[PHP File Handling](#)

## Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile, filesize("webdictionary.txt"));
fclose($myfile);
?>
```

Run example »

[PHP File Open/Read/Close](#)

## Example

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

[PHP File Create/Write](#)

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| basename()         | Returns the filename component of a path                            |
| chgrp()            | Changes the file group                                              |
| chmod()            | Changes the file mode                                               |
| chown()            | Changes the file owner                                              |
| clearstatcache()   | Clears the file status cache                                        |
| copy()             | Copies a file                                                       |
| delete()           | See unlink()                                                        |
| dirname()          | Returns the directory name component of a path                      |
| disk_free_space()  | Returns the free space of a filesystem or disk                      |
| disk_total_space() | Returns the total size of a filesystem or disk                      |
| diskfreespace()    | Alias of disk_free_space()                                          |
| fclose()           | Closes an open file                                                 |
| feof()             | Checks if the "end-of-file" (EOF) has been reached for an open file |

[PHP Filesystem Functions](#)

# Funkcje obsługi katalogów - katalogi

```
<?php
// bierzący katalog
$dir = "./";
// jeśli katalog istnieje otwórz bierzący katalog i pokaż zawartość
if (is_dir($dir)) {
    // pobierz deskryptor katalogu $dh
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            echo "nazwa: $file : typ: " . filetype($dir . $file) . "<br>";
        }
        // zamknij katalog
        closedir($dh);
    }
}
```

Zadanie:  
dokończyć katalogi

show.php

[opendir.php](#)

[PHP Directory Functions](#)

# Pliki cookies (ciasteczka)

Plik cookie to mały plik tekstowy, który serwer osadza na komputerze użytkownika. Za każdym razem, gdy przeglądarka zażąda strony internetowej z serwera, do żądania dodawane są pliki cookie należące do strony. W ten sposób serwer może zapisać informacje na komputerze użytkownika i mieć do nich dostęp.

Maksymalny rozmiar pliku cookie wynosi 4kB.

Pliki cookie zostały wprowadzone w celu rozwiązania problemu zapamiętania informacji o użytkowniku po zamknięciu przeglądarki.

Pliki cookie stosuje się w celu

- zarządzania sesjami - logowanie, zakupy online, wyniki gier itp..
- personalizacji – ustawienie preferencji użytkownika na stronie
- śledzenia (tracking) – śledzenie i analizowanie zachowania użytkownika w sieci

Pliki cookie nie są bezpieczne, gdyż są przechowywane otwartym tekstem po stronie użytkownika i każdy może je podejrzeć.

Pliki cookie można aktywować lub nie w przeglądarce.

Pliki cookie są zapisywane w formie par nazwa=wartość, np.:

username = Mirek



# Pliki cookies – funkcja setcookie()

funkcja setcookie() zapisuje ciasteczko w przeglądarce

tylko \$name jest wymagane, reszta parametrów jest opcjonalna

liczba sekund jakie upłyną od 1 stycznia 1970 roku, czyli od chwili zwanej początkiem epoki Uniksa (ang. Unix Epoch), systemu operacyjnego popularnego w latach 70 i 80. Jak upłynie ten czas cookie zostanie usunięte z przeglądarki.

```
setcookie(  
    string $name,           // nazwa  
    string $value = "",     // wartość  
    int    $expires = 0,    // czas trwania  
    string $path = "",      // ścieżka dostępu : "/" - dostępny w całej domenie,  
                             // "/www/pliki" - dostępny w podanym katalogu i podrzędnych  
    string $domain = "",    // domena, w której jest dostępny  
    bool   $secure = false, // jeśli true - może być przesyłany tylko przez szyfrowany protokół HTTPS  
    bool   $httponly = false // jeśli true - może być przesyłany tylko przez HTTP  
): bool  
    // wartość zwracana: true - sukces, false - błąd
```

parametry funkcji setcookie()

# Pliki cookies – skrypt zapisujący cookie w przeglądarce

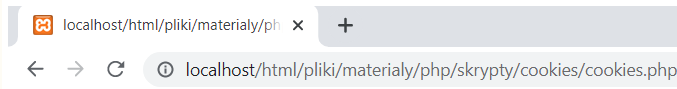
```
<?php
// wywołanie funkcji zapisującej ciasteczko w przeglądarce
$cookie_name = "user";
$cookie_value = "Jan Kowalski";
setcookie($cookie_name, $cookie_value, time() + 20, "/"); // 20s
?>

<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <h1>Cookies</h1>
    <?php
    // jeśli nie ma pliku cookie w przeglądarce
    if(!isset($_COOKIE[$cookie_name])) {
        echo "Cookie o nazwie '" . $cookie_name . "' nie istnieje!";
    }
    // jeśli ciasteczko istnieje wypisujemy jego nazwę i wartość
    else {
        echo "Cookie '" . $cookie_name . "' jest ustawione!<br>";
        echo "Przechowywana w cookie wartość: " . $_COOKIE[$cookie_name];
    }
    ?>
</body>
</html>
```

setCookies.php

ciasteczko będzie istniało 20s  
zwraca liczbę sekund jakie upłyną  
od 1 stycznia 1970 roku

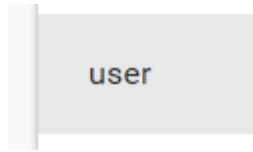
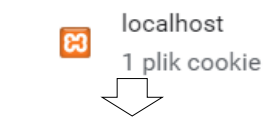
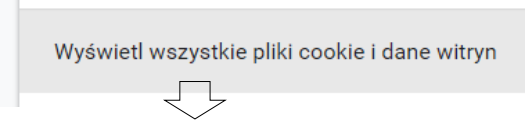
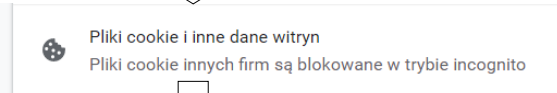
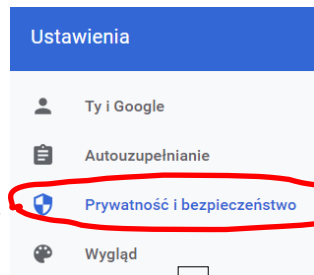
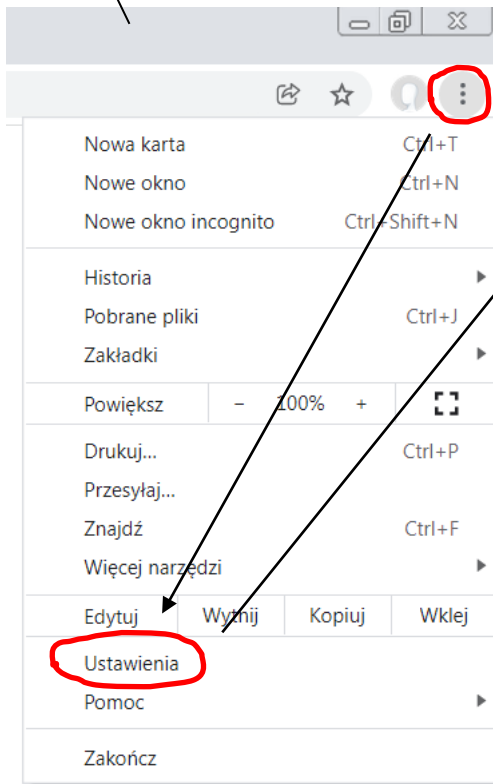
setcookie() definiuje plik cookie do wysłania wraz z resztą nagłówków HTTP. Podobnie jak inne nagłówki, ciasteczka muszą być wysyłane przed jakimkolwiek wyjściem ze skryptu (jest to ograniczenie protokołu). Wymaga to umieszczenia wywołań tej funkcji przed jakimkolwiek wyjściem, w tym znacznikami <html> i <head> oraz białymi znakami.



## Cookies

Cookie 'user' jest ustawione!  
Przechowywana w cookie wartość: Jan Kowalski

# Pliki cookies – sprawdzenie



user

Nazwa  
user

Treść  
Jan%20Kowalski

Domena  
localhost

Ścieżka  
/

Wyślij do:  
Tylko połączenia z tą samą witryną

Dostępny dla skryptu  
Tak

Utworzono  
piątek, 3 grudnia 2021 19:58:17

Wygasa  
piątek, 3 grudnia 2021 19:58:39

# Pliki cookies – skrypt usuwający cookie w przeglądarce

```
<?php
// wywołanie funkcji zapisującej ciasteczko w przeglądarce
$cookie_name = "user";
$cookie_value = "Jan Kowalski";
setcookie($cookie_name, $cookie_value, time() - 20, "/"); // ustawiamy czas, który upłynął
?>
```

```
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <h1>Cookies</h1>
```

aby usunąć cookie z przeglądarki musimy ustawić wsteczny czas, reszta parametrów powinna być taka sama jak przy zapisywaniu cookie (wartość cookie może być inna).

```
<?php
// jeśli nie ma pliku cookie w przeglądarce
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie o nazwie '" . $cookie_name . "' nie istnieje!";
}
// jeśli ciasteczko istnieje wypisujemy jego nazwę i wartość
else {
    echo "Cookie '" . $cookie_name . "' jest ustawione!<br>";
    echo "Przechowywana w cookie wartość: " . $_COOKIE[$cookie_name];
}
?>
```

```
</body>
</html>
```

# Pliki cookies – licznik odwiedzin

```
<?php
// jeśli nie istnieje w przeglądarce plik cookie 'licznik'
if (!isset($_COOKIE['licznik'])) {
    // ustawiamy ilość odwiedzin na 1
    $ilosc = 1;
}else{
    // gdy już jest w przeglądarce plik cookie 'licznik'
    // pobieramy z ciasteczka ilość odwiedzin
    // za pomocą funkcji intval(), która zwraca int
    $ilosc = intval($_COOKIE['licznik']);
    // inkrementujemy ilość odwiedzin
    $ilosc++;
}
// zapisujemy w przeglądarce plik cookie "licznik" z ilością odwiedzin $ilosc
// długość życia ciasteczka ustawiamy na 1 dzień
setcookie("licznik", $ilosc, time() + 60*60*24);
?>

<!doctype html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <h1>Licznik odwiedzin oparty o plik cookie</h1>
    <?php
    // wypisujemy na stronie ilość odwiedzin
    echo "licznik: $ilosc";
    ?>
</body>
</html>
```



**Licznik odwiedzin oparty o plik cookie**

licznik: 66

# Sesje

Po utworzeniu sesji można z niej korzystać na różnych stronach witryny i w różnych plikach na serwerze.

Sesja służy do zapisania informacji na serwerze.

Zmienne sesji są przechowywane po stronie serwera, po stronie klienta jest przechowywany identyfikator sesji (w pliku cookie lub przekazywany przez URL)

Sesja wygasa po zamknięciu witryny. Aby można było dalej korzystać z danych sesji należy je zapisać w bazie danych.

Sesja może przechowywać prawie nieograniczoną ilość danych.

Dane sesji są bezpieczne ponieważ są przechowywane na serwerze w formie zaszyfrowanej.



# Sesje

startujemy sesję, identyfikator sesji jest zapisany w pliku cookie i wysyłany do serwera z każdym żądaniem strony

```
<?php
session_start();
echo "identyfikator sesji: ". session_id();
?>
```



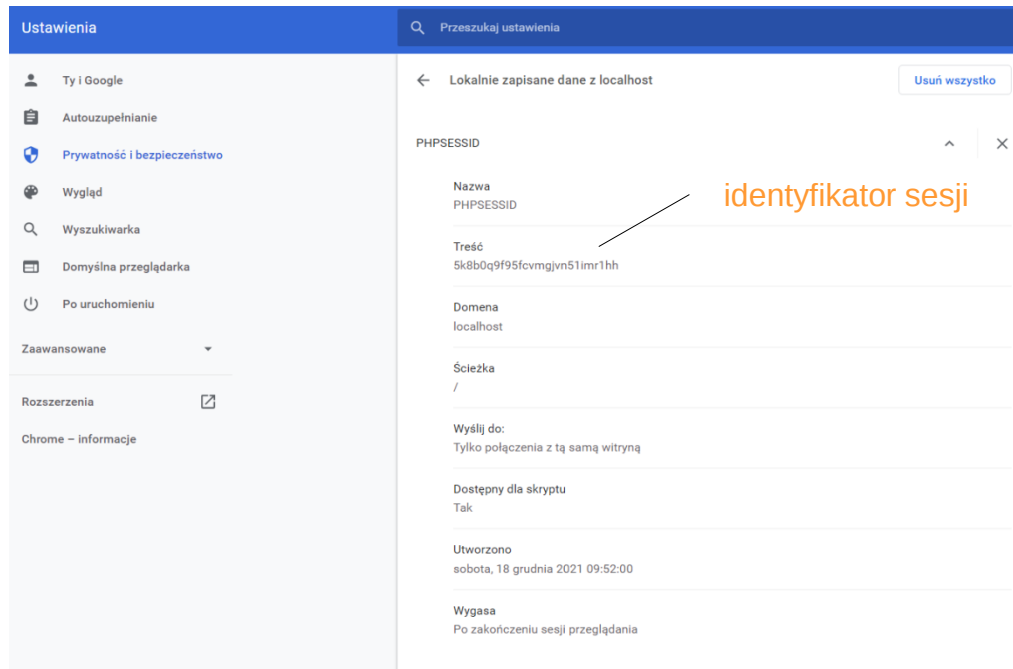
← → ↻ ⓘ localhost/html/pliki/materialy/php/

identyfikator sesji: 5k8b0q9f95fcvmgjn51imr1hh

identyfikator sesji

zmienne sesji przechowywane są na serwerze.

ustawienia przeglądarki Chrome



[sesje w3schools](#)

[funkcje związane z sesją](#)

# Sesje – przekazywanie zmiennych sesji między plikami

zmiennneSesji1.php

```
<?php
session_start();
$_SESSION["imie"] = "Jan";
$_SESSION["klasa"] = "IIb";
?>
<!doctype html>
<html>
<head>
  <title>zmiennne sesji</title>
  <meta charset="utf-8">
</head>
<body>
<h4>plik zmiennneSesji1.php</h4>
<?php
echo "<br>ustawiliśmy zmienne sesji:<br>";
echo "\$_SESSION["imie"]=" . $_SESSION["imie"] . "<br>";
echo "\$_SESSION["klasa"]=" . $_SESSION["klasa"] . "<br>";
?>
<a href="zmiennneSesji2.php">sprawdź zmienne sesji w drugim pliku</a>
</body>
</html>
```

startujemy sesję  
i tworzymy zmienne sesji



plik zmiennneSesji1.php

ustawiliśmy zmienne sesji:

`$_SESSION["imie"]=Jan`

`$_SESSION["klasa"]=IIb`

[sprawdź zmienne sesji w drugim pliku](#)

zmiennneSesji2.php

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
  <title>zmiennne sesji</title>
  <meta charset="utf-8">
</head>
<body>
<h4>plik zmiennneSesji2.php</h4>
<?php
echo "<br>zmiennne sesji odczytane lokalnie:<br>";
echo "\$_SESSION["imie"]=" . $_SESSION["imie"] . "<br>";
echo "\$_SESSION["klasa"]=" . $_SESSION["klasa"] . "<br>";
?>
<a href="zmiennneSesji1.php">powrót</a>
</body>
</html>
```

startujemy sesję  
odczytujemy zmienne sesji



plik zmiennneSesji2.php

zmienne sesji odczytane lokalnie:

`$_SESSION["imie"]=Jan`

`$_SESSION["klasa"]=IIb`

[powrót](#)

przekazywanie informacji  
w zmiennych sesji  
między plikami

# Sesje – licznik odwiedzin

```
<?php
session_start();
echo "identyfikator sesji: ". session_id();
if (!isset($_SESSION["odwiedziny"])){
    $_SESSION["odwiedziny"] = 1;    // tworzymy zmienną sesyjną odwiedzin
}
echo "<br>ilosc odwiedzin: ".$_SESSION['odwiedziny'];
$_SESSION["odwiedziny"]++;    // inkrementacja zmiennej sesyjnej odwiedzin

if ($_SESSION["odwiedziny"] == 10)
    session_destroy();    // usunięcie sesji po 9 odwiedzinach
?>
```

defaultowo zmienne sesji żyją aż do zamknięcia przeglądarki  
(jeśli chcemy informacje zatrzymać na dłużej należy użyć bazy danych)

identyfikator się nie zmienia

↓

identyfikator sesji: 5k8b0q9f95fcvmgjvn51imr1hh  
ilosc odwiedzin: 6

ilość się zmienia

ustawienia przeglądarki Chrome

↓

☒ Blokuj wszystkie pliki cookie (niezalecane)

↓

identyfikator się zmienia

identyfikator sesji: oleksfnrkns2blb64vk8lhbe17  
ilosc odwiedzin: 1

ilość się nie zmienia

# Sesje – logowanie

```
<?php
// startujemy sesję
session_start();
?>
<!doctype html>
<html>
<head>
    <title>logowanie</title>
    <meta charset="utf-8">
</head>
<body>
<h4>logowanie</h4>
<!-- formularz logowania -->
<form action="logowanie.php" method="post">
<p>Podaj swój login i hasło</p>
login: <input type="text" name="login">
<br><br>
hasło: <input type="password" name="hasło">
<br><br>
<input type="submit" value="zaloguj">
</form>
<p>
<?php
// w przypadku podania błędnego loginu lub hasła
// wyświetlamy komunikat o błędzie
if (isset($_SESSION["error"]))
    echo $_SESSION["error"];
else{
    unset($_SESSION["error"]);
}
?>
</p>
</body>
</html>
```

index.php

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
    <title>logowanie</title>
    <meta charset="utf-8">
</head>
<body>
<?php
// tworzymy zmienną sesyjną w celu powitania użytkownika
$_SESSION["powitanie"] = $_POST["login"];

// logowanie admina
if ($_POST["login"] == "admin" && $_POST["hasło"] == "admin"){
    // po podaniu poprawnego hasła i loginu przez admina tworzymy zmienną sesyjną
    // "adminZalogowany" w celu uwierzytelnienia admina w innych plikach
    $_SESSION["adminZalogowany"] = "sukces";
    header('location: admin.php');
    die();
}

// logowanie użytkownika
else if ($_POST["login"] == "adam" && $_POST["hasło"] == "777"){
    // po podaniu poprawnego hasła i loginu przez użytkownika
    // tworzymy zmienną sesyjną "zalogowany" w celu uwierzytelnienia użytkownika
    // w innych plikach
    $_SESSION["zalogowany"] = "sukces";
    header('location: zalogowany.php');
}

// błąd logowania
else{
    // w przypadku błędu logowania tworzymy zmienną sesyjną "error"
    // w celu wyświetlenia komunikatu w pliku index.php
    $_SESSION["error"] = "Error - nie jesteś zalogowany";
    header('location: index.php');
}
?>
</body>
</html>
```

logowanie.php

# Sesje – logowanie

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
    <title>zalogowany</title>
    <meta charset="utf-8">
</head>
<body>
<?php
// sprawdzamy czy zmienna sesyjna "zalogowany" jest ustawiona
// jeśli tak wyświetlamy rządą stronę
// jeśli nie odsyłamy do strony logowania
if (isset($_SESSION["zalogowany"]) && $_SESSION["zalogowany"]== "sukces"){
    echo "<h1>Witaj ".$_SESSION['powitanie']. "</h1><br><br>";
    echo "jesteś zalogowany<br>";
    echo "oto tajna treść:<br>";
    echo "Ala nie ma kota:<br>";
    echo "<a href='wyloguj.php'>wyloguj</a>";
}else{
    header('location: index.php');
}

?>
</body>
</html>
```

zalogowany.php

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
    <title>wyloguj</title>
    <meta charset="utf-8">
</head>
<body>
<?php
// usuniecie zmiennych sesji
session_unset();
// usunięcie sesji
session_destroy();
// powrót do strony logowania
header('location: index.php');
?>
</body>
</html>
```

wyloguj.php

# Sesje – logowanie

```
<?php
session_start();
?>
<!doctype html>
<html>
<head>
    <title>Admin</title>
    <meta charset="utf-8">
</head>
<body>
<?php
// sprawdzamy czy zmienna sesyjna "adminZalogowany" jest ustawiona
// jeśli tak wyświetlamy rządane stronę
// jeśli nie odsyłamy do strony logowania
if (isset($_SESSION["adminZalogowany"]) && $_SESSION["adminZalogowany"] == "sukces") {
echo "<h1>Witaj " . $_SESSION['powitanie'] . "</h1><br><br>";
echo "Informacje dla admina";
}
else{
    header('location: index.php');
}
?>

</body>
</html>
```

admin.php

# Sesje – logowanie

```
// jeśli login i hasło są poprawne przekierowujemy na stronę zalogowany.php
if ($_POST["login"] == "ala" && $_POST["haslo"] == "777"){
    // tworzymy zmienną sesyjną w celu uwierzytelnienia
    // zalogowanego użytkownika w innych plikach
    $_SESSION["user"] = "marek";
    header('location: zalogowany.php');
    exit();
}
```

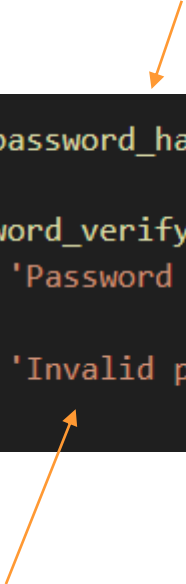


nigdy nie wpisujemy hasła bezpośrednio do skryptu!

login oraz zahaszowane hasło zwykle trzymamy w bazie danych

# Sesje – logowanie

funkcja haszująca – tworząca skrót hasła (unikalny ciąg znaków skojarzony z hasłem), defaultowo używa algorytmu bcrypt



```
$hash = password_hash("mojeHasło", PASSWORD_DEFAULT);

if (password_verify('mojeHasło', $hash)) {
    echo 'Password is valid!';
} else {
    echo 'Invalid password.';
}
```

weryfikacja hasła ze skrótem (haszem)

[password\\_hash](#)

[password\\_verify](#)

# Sesje – logowanie bcrypt

ciąg znaków wygenerowany przez funkcję haszującą `password_hash()` dla hasła `abc123xyz`

\$2a\$12\$R9h/cIPz0gi.URNNX3kh20PST9/PgBkqquzi.Ss7KIUg02t0jWMUw

Alg Cost

## Salt

## Hash

identyfikator  
algorytmu (bcrypt)

parametr związany z ilością  
przebiegów konfiguracji algorytmu  
– tutaj 12  
(czyim większa wartość tym dłużej  
algorytm haszuje hasło – żeby je  
złamać potrzeba więcej czasu)

dodatkowy,  
losowy ciąg  
znaków dodawany  
do haszu hasła

skrót hasła

## Bcrypt

### Cost

# Połączenie z bazą danych

```
<?php
$hostname = 'localhost';
$username = 'root';
$password = '';

// logowanie do bazy MySQL
$link = mysqli_connect($hostname, $username, $password);

// obsługa błędu połączenia z bazą
if (mysqli_connect_errno()) {
    printf("<br>Connect failed: %s\n<br>", mysqli_connect_error());
    exit();
} else {
    echo "połączenie nawiązane<br>";
}

// nazwa bazy danych
$db = "szkola";

// wybieramy bazę szkoła
if (mysqli_select_db($link, $db)){
    echo "baza danych $db wybrana</span><br>";
} else{
    die ("<br>Nie można wybrać bazy danych $db <br>");
}

// zapytanie do bazy: wybierz wszystkie kolumny z tabeli uzytkownik
$query="select * from uzytkownik";

// zapytanie przesyłamy do bazy i dostajemy odpowiedź $result
$result = mysqli_query($link,$query) or die("<br>Zapytanie niepoprawne<br>");

// pobieramy kolejne wiersze z odpowiedzi bazy
while ($line = mysqli_fetch_row($result)) {
    // id, login, password
    echo "<tr><td>".$line[0]."</td><td>".$line[1]."</td><td>".$line[2]."</td></tr>";
}

// zamykamy połączenie z bazą
mysqli_close($link);
?>
```

[Różne metody połączenia do bazy W3Schools](#)

# Połączenie z bazą danych

baza danych wedkowanie

kwerenda: wybierz wszystkie wiersze z tabeli ryby

The screenshot shows a MySQL database management interface. On the left, a tree view displays the database structure. The 'wedkowanie' database is selected, and the 'ryby' table is highlighted. An arrow points from the 'ryby' table to the query editor. The query editor contains the SQL query: `SELECT * FROM `ryby``. Below the query editor, there are options for 'Profilowanie' (Profiling) and 'Pokaż wszystko' (Show all), along with a 'Liczba wierszy' (Number of rows) dropdown set to 25 and a 'Filtrowanie wierszy' (Filtering rows) button. The results are displayed in a table with columns: id, nazwa, wystepowanie, and styl\_zycia. The table contains 8 rows of data. An arrow points from the text '1 oznacza ryby drapieżne' (1 indicates predatory fish) to the 'styl\_zycia' column, specifically to the value '1' in the first row.

Nowa

- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test
- wedkowanie
  - Nowy
  - lowisko
  - okres\_ochronny
  - ryby

```
SELECT * FROM `ryby`
```

☐ Profilowanie [ Edytuj w linii ] [ Edytuj ] [ Wyjaśnij SQL ] [ Utwórz kod PHP ] [ Odśwież ]

☐ Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przesz

+ Opcje

|                          |        |        |      | id | nazwa    | wystepowanie          | styl_zycia |
|--------------------------|--------|--------|------|----|----------|-----------------------|------------|
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 1  | Szczupak | stawy, rzeki          | 1          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 2  | Karp     | stawy, jeziora        | 2          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 3  | Sandacz  | stawy, jeziora, rzeki | 1          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 4  | Okon     | rzeki                 | 1          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 5  | Sum      | jeziora, rzeki        | 1          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 6  | Dorsz    | morza, oceany         | 1          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 7  | Leszcz   | jeziora               | 2          |
| <input type="checkbox"/> | Edytuj | Kopiuj | Usuń | 8  | Lin      | jeziora               | 2          |

1 oznacza ryby drapieżne

tabela ryby

wynik kwerendy

[MySQL tutorial](#)

# Połączenie z bazą danych

kwerenda: wybierz jedynie pola nazwa i wystepowanie z tabeli ryby dla ryb drapieżnych

```
SELECT nazwa, wystepowanie FROM `ryby` WHERE styl_zycia=1;
```

☐ Profilowanie [ Edytuj w linii ] [ Edytuj ] [ Wyjaśnij SQL ] [ Utwórz kod PHP ] [ Odśwież ]

☐ Pokaż wszystko | Liczba wierszy: 25 ▼ Filtrowanie wierszy: Przesz

+ Opcje

|                          |                                                                                   |        |                                                                                   |        | nazwa                                                                             | wystepowanie |          |                       |
|--------------------------|-----------------------------------------------------------------------------------|--------|-----------------------------------------------------------------------------------|--------|-----------------------------------------------------------------------------------|--------------|----------|-----------------------|
| <input type="checkbox"/> |  | Edytuj |  | Kopiuj |  | Usuń         | Szczupak | stawy, rzeki          |
| <input type="checkbox"/> |  | Edytuj |  | Kopiuj |  | Usuń         | Sandacz  | stawy, jeziora, rzeki |
| <input type="checkbox"/> |  | Edytuj |  | Kopiuj |  | Usuń         | Okon     | rzeki                 |
| <input type="checkbox"/> |  | Edytuj |  | Kopiuj |  | Usuń         | Sum      | jeziora, rzeki        |
| <input type="checkbox"/> |  | Edytuj |  | Kopiuj |  | Usuń         | Dorsz    | morza, oceany         |

wynik kwerendy

# Połączenie z bazą danych

```
<?php
echo "Połączenie z bazą danych<br><br>";

$link = mysqli_connect("localhost", "root", "");

if (mysqli_connect_errno())
    printf("Connect failed: %s\n", mysqli_connect_error());
else
    echo "połączenie nawiązane<br>";

if (mysqli_select_db($link, "wedkowanie"))
    echo "baza danych wedkowanie wybrana<br>";
else
    die ("Nie można wybrać bazy danych wedkowanie");

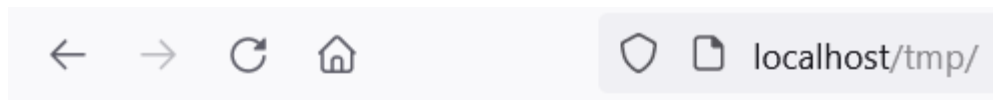
$query="SELECT nazwa, wystepowanie FROM `ryby` WHERE styl_zycia=1";

$result = mysqli_query($link,$query) or die("Zapytanie niepoprawne");

echo "<h2>Ryby drapieżne naszych wód</h2>";
echo "<ul>";
while ($line = mysqli_fetch_row($result)) {
    echo "<li>$line[0], występowanie: $line[1] </li>";
}
echo "</ul>";

mysqli_close($link);
?>
```

# Połączenie z bazą danych



Połączenie z bazą danych

połączenie nawiązane

baza danych wybrana

## Ryby drapieżne naszych wód

- Szczupak, występowanie: stawy, rzeki
- Sandacz, występowanie: stawy, jeziora, rzeki
- Okon, występowanie: rzeki
- Sum, występowanie: jeziora, rzeki
- Dorsz, występowanie: morza, oceany

egzamin

| <i>Efekt kształcenia</i>                            | <i>Kryterium weryfikacji</i>             |
|-----------------------------------------------------|------------------------------------------|
| Uczeń (zdający):                                    | Uczeń (zdający):                         |
| 4) programuje skrypty wykonywane po stronie serwera | 2) stosuje wbudowane instrukcje, funkcje |

**Przykładowe zadanie 21.**

```
function suma($n)
{
    if($n < 1) return 0;
    return $n + suma($n - 1);
}
echo suma(5);
```

Która wartość zostanie wyświetlona w wyniku działania skryptu PHP?

- A. 0
- B. 10
- C. 14
- D. 15

Odpowiedź prawidłowa: D

*Jednostka efektów kształcenia:*

**INF.03.5. Programowanie aplikacji internetowych**

| <i>Efekt kształcenia</i>                            | <i>Kryterium weryfikacji</i>             |
|-----------------------------------------------------|------------------------------------------|
| Uczeń (zdający):                                    | Uczeń (zdający):                         |
| 4) programuje skrypty wykonywane po stronie serwera | 2) stosuje wbudowane instrukcje, funkcje |

**Przykładowe zadanie 22.**

```
echo gettype(13.0);
```

Zapisana w PHP instrukcja spowoduje wypisanie typu

- A. string
- B. double
- C. integer
- D. boolean

**Odpowiedź prawidłowa: B**

# Źródła

<https://www.php.net/>