

Temat: Obsługa formularzy

Kryteria:

1. Walidacja

PHP – formularze

Podstawowym sposobem interakcji między użytkownikiem a serwisem WWW są formularze HTML.

Formularze HTML to specjalna konstrukcja języka HTML, dzięki której przeglądarka generuje zestaw elementów kontrolnych, pozwalających użytkownikowi na wprowadzanie danych i podejmowanie działań typu wybór elementu z listy rozwijanej, kliknięcie przycisku itp.

PHP –formularze

Obsługa formularzy wiąże się nie tylko z pobieraniem danych przekazywanych przez poszczególne pola formularza, ale także ze sprawdzaniem poprawności jego wypełnienia przez użytkownika.

Sprawdzanie poprawności/walidacja może dotyczyć różnych obszarów:

- ✓ **czy dane zostały wprowadzone,**
- ✓ **czy są odpowiedniego typu,**
- ✓ **czy spełniają dodatkowe ograniczenia.**

Dane przesyłane przez formularz są zawsze tekstem, mogą jednak zawierać tekst, który może być konwertowany na inny typ.

PHP –formularze, dane jako string

```
<form method="post">
<fieldset>
  <label for="name">Wprowadź swoje dane:</label><br><br>
  <input id="name" name="name" required><br><br>
  <label for="email">Email:</label>
  <input type="email" name="email" required><br><br>
  color: <input type="color" name="color" ><br><br>
  date: <input type="date" name="date" ><br><br>
  <label for="name">Podaj liczbę w zakresie od 1 do 30</label>
  <input type="number" name="number" min="1" max="30"><br><br>
  <input type="submit" name="send">
</fieldset>
</form>
```

```
<?php
if ( isset( $_POST['send'] ) ){
echo "<pre>";
var_dump($_POST);
echo "</pre>";
}

array(6) {
  ["name"]=>
  string(3) "ZSK"
  ["email"]=>
  string(13) "zsk@gmail.com"
  ["color"]=>
  string(7) "#394595"
  ["date"]=>
  string(10) "2018-10-11"
  ["number"]=>
  string(1) "3"
  ["send"]=>
  string(9) "Prześlij"
}
```

Wprowadź swoje dane:

ZSK

Email: zsk@gmail.com

color:

date: 11.10.2018

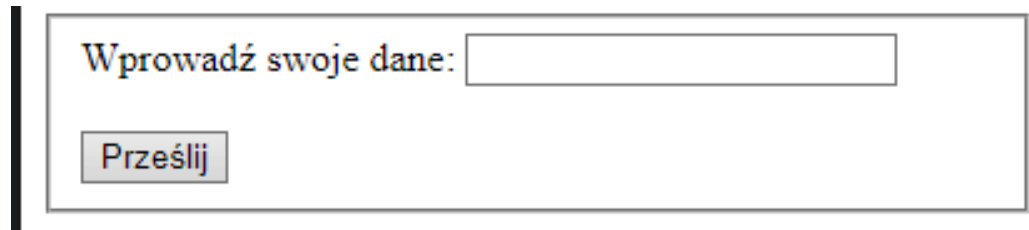
Podaj liczbę w zakresie od 1 do 30

3

Prześlij

PHP –formularze, walidacja tekstu

```
<form method="post" class="form1">
<fieldset>
    <label for="name">Wprowadź swoje dane:</label>
    <input id="name" name="name"><br><br>
    <input type="submit" name="send">
</fieldset>
</form>
```



Wprowadź swoje dane:

PHP – formularze, walidacja tekstu

```
<?php
// sprawdzamy, czy wysłano formularz
if ( isset( $_POST['send'] ) ){
    // sprawdzamy, czy wpisano dane
    if (empty($_POST["name"])){
        echo '<p style="color:red">Musisz wypełnić pole!</p>';
    } else {
        $text = trim($_POST["name"]);
        // sprawdzamy długość tekstu
        if (strlen($text) < 3) {
            echo '<p style="color:red">Text jest za krótki</p>';
        } else {
            echo "<h3>Witaj ".$text." </h3>";
        }
    }
}
```

Wprowadź swoje dane:

Musisz wypełnić pole!

Wprowadź swoje dane:

Text jest za krótki

Wprowadź swoje dane:

Witaj ZSK

PHP –formularze, walidacja liczb

Dane numeryczne

```
<form method="post" class="form1">
<fieldset>
    <label for="l_a">Podaj liczbę:</label>
    <input id="l_a" name="a"><br><br>
    <label for="l_b">Podaj drugą liczbę:</label>
    <input id="l_b" name="b"><br><br>
    <input type="submit" name="send" value='Oblicz średnią'>
</fieldset>
</form>
```

Podaj liczbę:

Podaj drugą liczbę:

Oblicz średnią

PHP – formularze, walidacja liczb

```
<?php
if ( isset( $_POST['send'] ) ){
    // sprawdzamy, czy wpisano dane
    if (empty($_POST["a"])|| empty($_POST["b"])){
        echo '<p style="color:red">Musisz podać obie liczby!</p>';
    } else {
        $a=$_POST["a"];
        $b=$_POST["b"];
        // sprawdzamy, czy podano liczby
        if(is_numeric($a)&&is_numeric($b)){
            $a=floatval($a);//zmiana typu danych
            $b=floatval($b);
            $srednia=($a+$b)/2;
            echo '<h3>Średnia dwóch liczb: '.$srednia.'</h3>';
        } else {
            echo '<p style="color:red">Nieprawidłowe dane!</p>';
        }
    }
}

?>
```

Podaj liczbę:

Podaj drugą liczbę:

Nieprawidłowe dane!

PHP –formularze

Zapamiętanie wpisanych danych w polach formularza

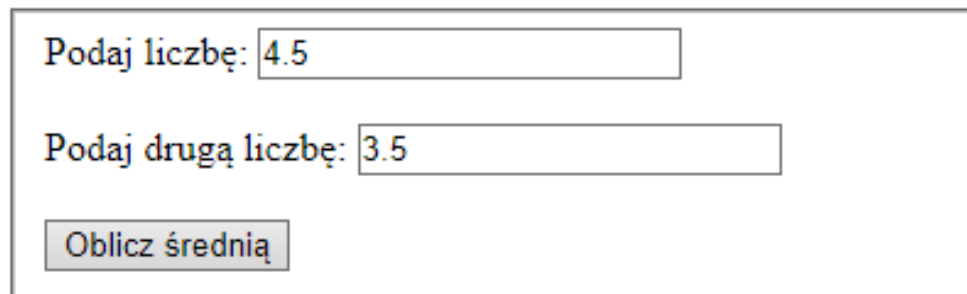
```
<?php
$a='';
$b='';
if ( isset( $_POST['send'] ) ){
    // sprawdzamy, czy wpisano dane
    if (empty($_POST["a"])|| empty($_POST["b"])){
        echo '<p style="color:red">Musisz podać obie liczby!</p>';
    } else {
        $a=$_POST["a"];
        $b=$_POST["b"];
        // sprawdzamy, czy podano liczby
        if(is_numeric($a)&&is_numeric($b)){
            $a=floatval($a);//zmiana typu danych
            $b=floatval($b);
            $srednia=($a+$b)/2;
        } else {
            echo '<p style="color:red">Nieprawidłowe dane!</p>';
        }
    }
}
?>
```

PHP –formularze

Zapamiętanie wpisanych danych w polach formularza

```
<form method="post" class="form1">
<fieldset>
    <label for="l_a">Podaj liczbę:</label>
    <input id="l_a" name="a" value='<?php echo $a;?>'><br><br>
    <label for="l_b">Podaj drugą liczbę:</label>
    <input id="l_b" name="b" value='<?php echo $b;?>'><br><br>
    <input type="submit" name="send" value='Oblicz średnią'>
</fieldset>
</form>

<?php
    if (isset($srednia)) {
        echo '<h3>Średnia dwóch liczb: '.$srednia.'</h3>';
    }
?>
```



Podaj liczbę:

Podaj drugą liczbę:

Średnia dwóch liczb: 4

PHP –formularze

Przesyłanie
danych do innej
strony

```
<?php
// Funkcja wyświetlająca formularz
function formularz() {
?>
<form class="form1" action="l9p5.php" method="post">
  <fieldset>
  <div>
    <label for="name1">Imię:</label><br>
    <input name="imie" id="name1" value=""><br>
    <label for="name2">Nazwisko:</label><br>
    <input name="nazwisko" id="name2" value=""><br>
    <label for="profession">Zawód:</label><br>
    <input name="zawod" id="profession" value=""><br>
    <label for="email">Adres e-mail:</label> <br>
    <input name="email" id="email" value=""><br>
    <input type="checkbox" name="mailing" id="mailing" value="checked">
    <label for="mailing">Chcę otrzymywać informacje handlowe:</label><br><br>
    <input type="submit" value="Wyślij" name="submit">
    <input type="reset" value="Wyczyść" name="reset">
  </div>
  </fieldset>
</form>
<?php
}
?>
<?php
formularz();
?>
```

http://localhost/zsk/l9p5_form.php

PHP – formularze

Przesyłanie danych do innej strony; cdn.

```
8  <?php
9  if (isset($_POST["submit"])) {
10 // - sprawdzamy, czy wszystkie pola zostały wypełnione
11 if (empty($_POST["imie"]) ||
12     empty($_POST["nazwisko"]) ||
13     empty($_POST["zawod"]) || empty($_POST["email"])) {
14     echo '<p style="color:red">Musisz wypełnić wszystkie pola!</p>';
15     echo '<p><a href="l9p5_form.php">Powrót do formularza</a></p>';
16 } else {
17     ?>
18     <h3>Dziękujemy za zgłoszenie!</h3>
19     <p>Twoje dane:
20     <ul>
21     <li>Imię: <b><?php echo trim($_POST["imie"]); ?></b></li>
22     <li>Nazwisko: <b><?php echo trim($_POST["nazwisko"]); ?></b></li>
23     <li>Zawód: <b><?php echo trim($_POST["zawod"]); ?></b></li>
24     <li>Adres email: <b><?php echo trim($_POST["email"]); ?></b></li>
25     <?php
26     if (isset($_POST["mailing"])){
27     echo "<li>Chcesz otrzymywać informacje handlowe.</li>";
28     } else {
29     echo "<li>Nie chcesz otrzymywać informacji handlowych.</li>";
30     }
31     ?>
```

<http://localhost/zsk/l9p5.php>

PHP –formularze

Przesyłanie danych do innej strony;

```
32         </ul>
33     </p>
34     <?php
35     }
36     } else {
37     /* Jeśli użytkownik dostał się na tę stronę
38     w sposób inny niż przez formularz
39     zostaje przekierowany do formularza zgłoszenia*/
40     header("Location: 19p5_form.php");
41     }
42     ?>
```

PHP –formularze

Informacje o błędzie
przy poszczególnych
polach;

```
1 <?php
2 $email = '';
3 $password = '';
4 $terms = '';
5 $errorEmail = '';
6 $errorPassword = '';
7 $errorTerms = '';
8 if ( isset( $_POST['send'] ) ) {
9     $email = $_POST['email'];
10    $password = $_POST['password'];
11    if (isset($_POST['terms'])) {
12        $terms = $_POST['terms'];
13    }
14    if ( ! $email ) {
15        $errorEmail = 'Uzupełnij pole email';
16    }
17    if ( ! $password ) {
18        $errorPassword = 'Uzupełnij pole hasło';
19    }
20    if ( $terms != 'on' ) {
21        $errorTerms = 'Musisz zaakceptować regulamin';
22    }
23 }
24 ?>
```

PHP –formularze

Informacje o błędzie przy poszczególnych polach;

```
<h4>Rejestracja</h4>
<div class="form1">
  <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <fieldset>
      <div class="field">
        <label for="email">Email (login)</label>
        <?php if ( $errorMessage != null ) { ?>
          <span class="red_label">
            <?php echo $errorMessage; ?>
          </span>
        <?php } ?>
        <input type="text" name="email" id="email">
      </div>
      <div class="field">
        <label for="password">Hasło</label>
        <?php if ( $errorMessage != null ) { ?>
          <span class="red_label">
            <?php echo $errorMessage; ?>
          </span>
        <?php } ?>
        <input type="password" name="password" id="password">
      </div>
```

<http://localhost/zsk/l9p6.php>

PHP –formularze

Informacje o błędzie przy poszczególnych polach;

```
<?php if ( $errorTerms != null ) { ?>
    <span class="red_label">
        <?php echo $errorTerms; ?>
    </span>
<?php } ?>
<div class="field2">
<input type="checkbox" tabindex="0" name="terms" id="terms">
<label for="terms">Zapoznałem się z regulaminem</label>
</div>
</div>
<input type="submit" id="send" name="send" value="Wyślij">
</fieldset>
</form>
```

Rejestracja

Email (login)

Uzupełnij pole email

Hasło

Uzupełnij pole hasło

Musisz zaakceptować regulamin

☐ Zapoznałem się z regulaminem

Wyślij

<http://localhost/zsk/l9p6.php>

PHP –formularze, dodatkowe ograniczenia

```
<?php
    $a=$b="";
    $blad_a=$blad_b="";
    $a_ok=$b_ok=false;
    $start=' <p class="red_label">';
    $end='</p>';
    if ( isset( $_POST['send'] ) ){
        if(isset($_POST["a"])){
            $a=$_POST["a"];
            if($a==""){
                $blad_a="Pole nie może być puste";
            } elseif(!is_numeric($a)){
                $blad_a="Niepoprawny format liczby";
            } elseif($a<=0){
                $blad_a="Długość boku musi być dodatnia";
            } else{
                $a_ok=true;
            }
        }
    }
```

PHP –formularze, dodatkowe ograniczenia

```
if(isset($_POST["b"]))
{
    $b=$_POST["b"];
    if($b=="") {
        $blad_b="Pole nie może być puste";
    } elseif(!is_numeric($b)) {
        $blad_b="Niepoprawny format liczby";
    } elseif($b<=0) {
        $blad_b="Długość boku musi być dodatnia";
    } else {
        $b_ok=true;
    }
}
```

PHP – formularze, dodatkowe ograniczenia

```
<form method="post" class="form1">
<fieldset>
    <label for="l_a">Podaj bok a:</label>
    <input id="l_a" name="a" value='<?php echo $a;?>'>
    <?php echo $start.$blad_a.$end;?><br>
    <label for="l_b">Podaj bok b:</label>
    <input id="l_b" name="b" value='<?php echo $b;?>'>
    <?php echo $start.$blad_b.$end;?><br><br>
    <input type="submit" name="send" value='Oblicz obwód i pole prostokąta'>
</fieldset>
</form>

<?php
    if($a_ok&&$b_ok)
    {
        echo "Pole prostokąta wynosi: " . ($a*$b) .
        "<br>Obwód prostokąta wynosi: " . (2*($a+$b)) ;
    }

    ?>
```

Podaj bok a: Długość boku musi być dodatnia

Podaj bok b: Niepoprawny format liczby

PHP –formularze, email

Sprawdzenie poprawności adresu email:

- ✓ podstawowa walidacja HTML 5 pola email
- ✓ wykorzystanie wyrażeń regularnych
- ✓ wykorzystanie wbudowanych filtrów

PHP –formularze, wyrażenia regularne

Wyrażenia regularne (z ang. regular expressions) to mechanizm obsługi wzorców, pozwalający odnajdywać i weryfikować czy podany tekst odpowiada zdefiniowanemu przez programistę warunkom (tj. podanemu wzorcowi). Przykładowym zastosowaniem wyrażen regularnych jest weryfikowanie poprawności danych przesyłanych przez użytkownika np. czy wpisany adres e-mail jest poprawny.

Funkcja

preg_match (wzorzec, ciąg)

wyszukuje wzorzec w ciągu, zwracając wartość true, jeśli wzorzec istnieje, i wartość false w przeciwnym razie.

PHP –formularze, wyrażenia regularne

Regeksy są dość skomplikowane, dlatego tutaj tylko kilka przykładów ich zastosowania:

abc- znak a, po którym następuje b, a następnie c.

a*- dopasowuje znak a, zero lub więcej razy

a + - dopasowuje znak a jeden lub więcej razy

[^a]- dopasowuje jeden znak, który nie jest a.

[abc] - dopasowuje jeden znak: a, b lub c.

[a-z]- dopasowuje dowolny znak w zakresie a-z, tylko małe litery

[A-Za-z] - małe i wielkie litery

a.c - a, następnie dowolny znak, później c.

a{5} - dopasuje a, 5 razy.

a{5,7} - dopasuje a, 5 do 7 razy, ale nie mniej ni więcej.

PHP –formularze, wyrażenia regularne

[-] - dopasowuje spację lub myślnik.

[0-9] - dopasowuje dowolną cyfrę z zakresu od 0 do 9.

[0-9]{3}[-][0-9]{3}[-][0-9]{3}- test numeru telefonu - trzy cyfry, po którym następuje spacja lub myślnik, a następnie trzy cyfry, po których następuje spacja lub myślnik, i znów trzy cyfry.

[0-9]{2}[-][0-9]{3} – test kodu pocztowego, dwie cyfry, po którym następuje myślnik, a następnie trzy cyfry

Imię i nazwisko:

[A-Z][a-z]+[] [A-Z][a-z]+ - Imię z dużej litery [A-Z], dowolnej (min. 1)ilości małych liter[a-z]+ , spacji [] i nazwisko z dużej litery i dowolnej ilości małych [A-Z][a-z]+

PHP –formularze, wyrażenia regularne

adres meilowy:

`^[0-9a-zA-Z_.-]+@[0-9a-zA-Z_.-]+\.[a-zA-Z]{2,3}$`

^ - Wzorzec ma się zaczynać z początkiem tekstu

\$ - Wzorzec ma się kończyć z końcem tekstu

[0-9a-zA-Z_.-]+ - badamy nazwę konta, która może składać się z dowolnych znaków (cyfry, litery, .-_)

@ - sprawdzamy wystąpienie znaku @

[0-9a-zA-Z_.-]+ - po znaku @ sprawdzamy domenę, która może składać się z takich samych znaków co nazwa konta oprócz znaku _

\. - po domenie musi wystąpić kropka

[a-zA-Z]{2,3} - po kropce musi wystąpić końcówka domeny, która może się składać wyłącznie z liter i jej długość musi być od 2 do 3 znaków

PHP –formularze, email

```
<?php
$email="" ;
$email_err="";
$regex="/^[0-9a-zA-Z_.-]+@[0-9a-zA-Z.-]+\.[a-zA-Z]{2,3}$/";
if ( isset( $_POST['send'] ) ){
    if( isset( $_POST['email'] ) ){
        $email = trim($_POST['email']);
        if (!preg_match($regex, $email)){
            $email_err= '<p class="red_label">email jest niepoprawny</p>';
        }
    }
}
```

```
?>
<form method="post" class="form1">
<fieldset>
    <label for="email">Email:</label>
    <input type="text" name="email" value='<?php echo $email;?>' required>
    <?php echo $email_err;?>
    <br><br>
    <input type="submit" name="send" value="wyślij">
</fieldset>
</form>
```

Email: email jest niepoprawny

PHP –formularze, regex

```
<?php
    $sname="";
    $sname_err="";
    $fname="";
    $fname_err="";
    $kod_post="";
    $kod_post_err="";
    $regex1="/^[A-Z][a-z]+$/";
    $regex2="/^[0-9]{2}[-][0-9]{3}$/";
    if ( isset( $_POST['send'] ) ){
        if( isset( $_POST['sname'] ) ){
            $sname = trim($_POST['sname']);
            if (!preg_match($regex1, $sname)){
                $sname_err= '<p class="red_label">Nazwisko
                powinno zaczynać się wielką literą</p>';
            }
        }
    }
```

PHP –formularze, regex

```
}  
if( isset( $_POST['fname'] ) ){  
    $fname = trim($_POST['fname']);  
    if (!preg_match($regex1, $fname)){  
        $fname_err= '<p class="red_label">Imię  
        powinno zaczynać się wielką literą</p>';  
    }  
}  
if( isset( $_POST['kod_post'] ) ){  
    $kod_post = trim($_POST['kod_post']);  
    if (!preg_match($regex2, $kod_post)){  
        $kod_post_err= '<p class="red_label">Podaj  
        kod w formacie np. 00-325</p>';  
    }  
}  
}
```

?>

PHP –formularze, regex

```
<form method="post" action="" class="form1">
<fieldset>
  <label for="sname">Nazwisko:</label>
  <input type="text" name="sname" value='<?php echo $sname;?>' required>
  <?php echo $sname_err;?>
  <br><br>
  <label for="fname">Imię:</label>
  <input type="text" name="fname" value='<?php echo $fname;?>' required>
  <?php echo $fname_err;?>
  <br><br>
  <label for="kod_post">Kod pocztowy:</label>
  <input type="text" name="kod_post" value='<?php echo $kod_post;?>' required>
  <?php echo $kod_post_err;?>
  <br><br>
  <input type="submit" name="send" value="wyślij">
</fieldset>
</form>
```

Nazwisko: Nazwisko powinno zaczynać się wielką literą

Imię:

Kod pocztowy: Podaj kod w formacie np. 00-325

PHP –formularze, filtry

Wykorzystanie wbudowanych filtrów

PHP oferuje kilka przydatnych do walidacji filtrów, np.:

`FILTER_VALIDATE_EMAIL`

`FILTER_VALIDATE_FLOAT`

`FILTER_VALIDATE_INT`

`FILTER_VALIDATE_URL` i inne.

Wykorzystując funkcję

`filter_var(zmienna, filtr)`

możemy w szybki sposób dokonać walidacji pól formularza. Sposób ten ma jednak pewne ograniczenia i nie daje gwarancji pełnej poprawności walidacji.

PHP –formularze, filtry

```
<?php
$email="";
$email_err="";
if ( isset( $_POST['send'] ) ){
    if( isset( $_POST['email'] ) ){
        $email = trim($_POST['email']);
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)){
            $email_err= '<p class="red_label">email jest niepoprawny</p>';
        }
    }
}

?>

<form method="post" class="form1">
<fieldset>
    <label for="email">Email:</label>
    <input type="text" name="email" value='<?php echo $email;?>' required>
    <?php echo $email_err;?>
    <br><br>
    <input type="submit" name="send" value="wyślij">
</fieldset>
</form>
```



The screenshot shows a web browser window displaying a form. The form has a label "Email:" followed by a text input field containing the value "zsk". To the right of the input field, there is a red error message: "email jest niepoprawny". Below the input field, there is a button labeled "wyślij".

PHP –formularze, filtry

```
1 <?php
2 $email = '';
3 $password = '';
4 $terms = '';
5 $errorEmail = '';
6 $errorPassword = '';
7 $errorTerms = '';
8 if ( isset( $_POST['send'] ) ) {
9     $email = $_POST['email'];
10    $password = $_POST['password'];
11    if (isset($_POST['terms'])) {
12        $terms = $_POST['terms'];
13    }
14    if ( ! $email ) {
15        $errorEmail = 'Uzupełnij pole email';
16    } elseif ( $email && !filter_var($email, FILTER_VALIDATE_EMAIL) ) {
17        $errorEmail = 'Upewnij się, że adres email ma prawidłowy format';
18    }
19    if ( ! $password ) {
20        $errorPassword = 'Uzupełnij pole hasło';
21    } elseif ( $password && strlen($password)<6 ) {
22        $errorPassword = 'Hasło musi zawierać minimum 6 znaków';
23    }
24    if ( $terms != 'on' ) {
25        $errorTerms = 'Musisz zaakceptować regulamin';
26    }
27 }
```


PHP –formularze, filtry

Rejestracja

Email (login)
Upewnij się, że adres email ma prawidłowy format
zsk@gmailcom

Hasło
Hasło musi zawierać minimum 6 znaków
...

Musisz zaakceptować regulamin
☐ Zapoznałem się z regulaminem

Wyślij

PHP –formularze, filtry

```
<?php
$a='';
$b='';
$srednia='';
if ( isset( $_POST['send'] ) ){
    // sprawdzamy, czy wpisano dane
    if (empty($_POST["a"])|| empty($_POST["b"])){
        echo '<p style="color:red">Musisz podać obie liczby!</p>';
    } else {
        $a=$_POST["a"];
        $b=$_POST["b"];
        if(filter_var($a,FILTER_VALIDATE_INT) !==false &&
        filter_var($b,FILTER_VALIDATE_INT) !==false){
            echo '<p style="color:red">Podałeś liczby całkowite</p>';
            $srednia=($a+$b)/2;
        }
        else {
            echo '<p style="color:red">Nieprawidłowe dane!</p>';
        }
    }
}
```

?>

PHP –formularze, filtry

```
<form method="post" class="form1">
<fieldset>
    <label for="l_a">Podaj liczbę całkowitą:</label>
    <input id="l_a" name="a" value='<?php echo $a;?>'><br><br>
    <label for="l_b">Podaj drugą liczbę całkowitą:</label>
    <input id="l_b" name="b" value='<?php echo $b;?>'><br><br>
    <input type="submit" name="send" value='Oblicz średnią'>
</fieldset>
</form>
<?php
    if (!empty($srednia)) {
        echo '<h3>Średnia dwóch liczb: ' . $srednia . '</h3>';
    }
?>
```

Nieprawidłowe dane!

Podaj liczbę całkowitą:

3

Podaj drugą liczbę całkowitą:

4.5

Oblicz średnią

rozwinięcie przykładu l9p4.php

<http://localhost/zsk/l9p11.php>

PHP –formularze,

Zapamiętanie wartości w polach radio;

```
<?php
$imie=$plec="";
$error_imie="";
$imie_ok=false;
$plec="";
$error_plec="";
$plec_ok=false;
if(isset($_POST["post"])){
    if(isset($_POST["imie"]))
    {
        $imie=$_POST["imie"];
        if($imie=="")
        {
            $error_imie="Pole nie może być puste";
        }
        else
        {
            $imie_ok=true;
        }
    }
    if(empty($_POST["plec"])){
        $error_plec="Określ płeć";
    }
    else{
        $plec=$_POST["plec"];
        $plec_ok=true;
    }
}
```

<http://localhost/zsk/l9p12.php>

PHP – formularze

```
<form action='' method='POST'>
  <input type='text' name='imie' value='<?php echo $imie;?>'>
  <?php echo "<span class='red_label'>". $error_imie. "</span>";?></br>
  Płeć:
  <input type='radio' id='m' name='plec' value='meczczyna'
  <?php if($plec_ok=true && $plec=="meczczyna") echo "checked";?>>
  <label for='m'>Mężczyzna</label>
  <input type='radio' id='k' name='plec' value='kobieta'
  <?php if($plec_ok=true && $plec=="kobieta") echo "checked";?>>
  <label for='k'>Kobieta</label>
  <?php echo "<span class='red_label'>". $error_plec. "</span>";?></br>
  <input type='submit' name='post' value='Wyślij'>
</form>
<?php
if($imie_ok)
{
    echo "Witaj ".$imie."! ";
    if($plec_ok=true && $plec=="kobieta") echo "Miło Panią poznać.";
    if($plec_ok=true && $plec=="meczczyna") echo "Miło Pana poznać.";
}
```

Pole nie może być puste

Płeć: ☐ Mężczyzna ☐ Kobieta **Określ płeć**

Płeć: ☒ Mężczyzna ☐ Kobieta

Witaj ZSK! Miło Pana poznać.

Kryteria:

2. Bezpieczeństwo

PHP – formularze, bezpieczeństwo

```
<form method="post" action="" class="form1">
<fieldset>
    <label for="sname">Nazwisko:</label>
    <input type="text" name="sname" value=''>
    <br><br>
    <label for="fname">Imię:</label>
    <input type="text" name="fname" value=''>
    <br><br>
    <label for="kod_post">Kod pocztowy:</label>
    <input type="text" name="kod_post" value=''>
    <br><br>
    <input type="submit" name="send" value="wyślij">
</fieldset>
</form>
```

```
<?php
    if ( isset( $_POST['send'] ) ){
        if ( isset( $_POST['sname'] ) ){
            echo    $_POST['sname'];
        }
        if ( isset( $_POST['fname'] ) ){
            echo    $_POST['fname'];
        }
    }
}
```

<?>

Nazwisko:

Imię:

Kod pocztowy:

I co mi zrobisz?

PHP –formularze, bezpieczeństwo

Podczas przetwarzania formularzy PHP należy pamiętać o **BEZPIECZEŃSTWIE**.

Aby chronić formularz przed hakerami i spamerami niezbędna jest prawidłowa weryfikacja danych przesyłanych z formularzy.

Fakt, że pola formularzy przesyłają wartości w postaci tekstu, umożliwia wpisanie w nie także kodu HTML czy JavaScript. To stanowi potencjalne niebezpieczeństwo i umożliwia różnego rodzaju ataki:

- ✓ **cross-site scripting (XSS)**
- ✓ **SQL injection**
- ✓ **Cross-site request forgery (CSRF) i inne**

PHP –formularze, bezpieczeństwo

Cross-site scripting (XSS)

to rodzaj luki w zabezpieczeniach komputera, zwykle występującej w aplikacjach internetowych. XSS umożliwia intruzom wstrzykiwanie skryptu po stronie klienta na strony internetowe przeglądane przez innych użytkowników. Atakujący ma możliwość wykonania dowolnego kodu skryptowego w przeglądarce (nie mylić z uruchomieniem dowolnego kodu w systemie operacyjnym ofiary).

Więcej: <https://sekurak.pl/czym-jest-xss/>

PHP –formularze, bezpieczeństwo

SQL injection

To jedna z dość częstych i jednocześnie niebezpiecznych podatności w aplikacjach webowych (oraz niewebowych). Atakujący wstrzykuje do aplikacji (nieautoryzowany) fragment zapytania SQL. Wstrzyknięcie zazwyczaj możliwe jest z jednego powodu – **braku odpowiedniego sprawdzenia (walidacji) parametru przekazanego przez użytkownika**. Taki parametr, gdy mamy do czynienia z SQL injection, często przekazywany jest **bezpośrednio** do zapytania SQL.

Więcej: <https://sekurak.pl/czym-jest-sql-injection/>

PHP –formularze, bezpieczeństwo

Cross-site request forgery (CSRF)

Bardzo niebezpieczny typ ataku, polegający na wykorzystywaniu uprawnień osoby atakowanej. Polega on na przesłaniu osobie posiadającej uprawnienia do wykonania danej akcji (najczęściej administrator), linka który daną akcję wykonuje

Więcej: <https://sekurak.pl/czym-jest-podatnosc-csrf-cross-site-request-forgery/>

PHP –formularze, bezpieczeństwo

Filtrowanie Danych

Wszystkie dane przesłane do skryptu przez użytkowników powinny być dokładnie filtrowane pod kątem obecności w nich kodu HTML i PHP, do tego celu należy używać funkcji:

- ✓ **htmlspecialchars()**

- ✓ **strip_tags()**

Warto także używać funkcji (odpowiednio dodającej i usuwającej znak \ przed znakami: ' , " , / , NULL) :

- ✓ **addslashes()** - przed zapisaniem tekstu do bazy/pliku

- ✓ **stripslashes()** - przed wyświetleniem tekstu z bazy/pliku

PHP –formularze, bezpieczeństwo

Funkcja

htmlspecialchars ()

konwertuje znaki specjalne na encje HTML.

Oznacza to, że zastąpi znaki HTML, takie jak np. <i>, znakami & lt; i & gt ;.

Zapobiega to wykorzystywaniu kodu przez intruza poprzez wstrzykiwanie kodu HTML lub kodu JavaScript (ataki typu Cross-site Scripting) w formularzach.

PHP –formularze, bezpieczeństwo

Funkcja

strip_tags(\$str [, \$allowable_tags])

Usuwa znaczniki HTML i PHP z łańcucha

Możesz użyć opcjonalnego drugiego parametru, aby określić znaczniki, które nie powinny być usunięte.

Ponieważ strip_tags () faktycznie nie sprawdza poprawności kodu HTML, częściowe lub uszkodzone znaczniki mogą spowodować usunięcie większej ilości tekstu / danych niż oczekiwano.

PHP –formularze, bezpieczeństwo

```
<?php
if ( isset( $_POST['send'] ) ){
    if ( isset( $_POST['sname'] ) ){
        echo "<h4>htmlspecialchars:</h4>";
        echo    htmlspecialchars($_POST['sname'])."<br>";
    }
    if ( isset( $_POST['fname'] ) ){
        echo    htmlspecialchars($_POST['fname'])."<br>";
    }
    if ( isset( $_POST['sname'] ) ){
        echo "<h4>strip_tags:</h4>";
        echo    strip_tags($_POST['sname'])."<br>";
    }
    if ( isset( $_POST['fname'] ) ){
        echo    strip_tags($_POST['fname'])."<br>";
    }
}
?>
```

htmlspecialchars:

<p style="color:red;">I co mi zrobisz?</p>
<script>alert('A teraz?')</script>

strip_tags:

I co mi zrobisz?
alert('A teraz?')

PHP –formularze, bezpieczeństwo

Dla każdej zmiennej przekazanej przez formularz warto wykonać następujące czynności:

- ✓ Usunąć niepotrzebne znaki (dodatkowe spacje, tabulacje, znaki nowej linii) z danych wejściowych użytkownika (za pomocą funkcji PHP `trim()`)
- ✓ Usunąć ukośniki odwrotne (`\`) z danych wejściowych użytkownika (za pomocą funkcji PHP `stripslashes()`)
- ✓ Konwertuj znaki specjalne na encje HTML za pomocą funkcji `htmlspecialchars()`

Dobrym sposobem jest przygotowanie funkcji wykonującej wszystkie te czynności.

PHP – formularze, bezpieczeństwo

```
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
```

?>

```
<?php
if ( isset( $_POST['send'] ) ){
    if ( isset( $_POST['sname'] ) ){
        echo "<h4>Funkcja czyszcząca:</h4>";
        echo test_input($_POST['sname'])."<br>";
    }
    if ( isset( $_POST['fname'] ) ){
        echo test_input($_POST['fname'])."<br>";
    }
    if ( isset( $_POST['kod_post'] ) ){
        echo test_input($_POST['kod_post'])."<br>";
    }
}
```

?>

Nazwisko:

Imię:

Kod pocztowy:

wyślij

Funkcja czyszcząca:

<p style="color:red;">I co mi zrobisz?</p>
<script>alert('A teraz?')</script>
\\44532

PHP –formularze, podsumowanie

```
1  <!DOCTYPE HTML>
2  <html lang="pl">
3  <head>
4      <meta charset="utf-8"/>
5      <title>Przetwarzanie formularzy -l9p16_form</title>
6      <style>
7          .form1{
8              width:30vw;
9          }
10         .red_label
11         {
12             display:inline-block;
13             background-color:#B60C1B;
14             font-size:12px;
15             color: #fff;
16         }
17         .field{
18             display:flex;
19             justify-content:flex-start;
20             flex-direction: column;
21             padding: 5px;
22         }
23     </style>
24 </head>
```

http://localhost/zsk/l9p16_form.php

PHP –formularze, podsumowanie

```
<body>
<h3>Przetwarzanie formularzy</h3>
<div class="form1">
<form action="" method="post" id="form">
  <fieldset>
    <div class="field">
      <label for="name">Imię: <i>*</i></label>
      <input type="text" name="name" id="name" class="req" value="<?php echo $form['name']; ?>" />
      <?php echo $error['name']; ?>

      <label for="phone">Telefon: <i>*</i></label>
      <input type="text" name="phone" id="phone" class="req" value="<?php echo $form['phone']; ?>" />
      <?php echo $error['phone']; ?>

      <label for="email">Email: <i>*</i></label>
      <input type="text" name="email" id="email" class="req" value="<?php echo $form['email']; ?>" />
      <?php echo $error['email']; ?>

      <label for="message">Wiadomość: </label>
      <textarea name="message" id="message"><?php echo $form['message']; ?></textarea>

      <p>*</p>

      <input type="submit" name="submit" id="submit" value="Wyślij"/>
    </div>
  </fieldset>
</form>
</div>
</body>
</html>
```

PHP –formularze, podsumowanie

```
1 <?php
2 $error_start = "<p class='red_label'>";
3 $error_end = "</p>";
4 $valid_form = TRUE;
5 $regex='/^[0-9a-zA-Z_.-]+@[0-9a-zA-Z.-]+\.[a-zA-Z]{2,3}$/';
6 $form_fields = array('name','phone','email','message');
7 $required = array('name','phone','email');
8
9 foreach($required as $require){
10     $error[$require] = '';
11 }
12
13 if(isset($_POST['submit'])){
14     foreach($form_fields as $field){
15         $form[$field] = htmlspecialchars($_POST[$field]);
16     }
17
18     if($form['name'] == ''){
19         $error['name'] = $error_start . "Wypełnij wymagane pole" . $error_end;
20         $valid_form = FALSE;
21     }
22
23     if($form['phone'] == ''){
24         $error['phone'] = $error_start . "Wypełnij wymagane pole" . $error_end;
25         $valid_form = FALSE;
26     }
```

PHP –formularze, podsumowanie

```
27
28 if($form['email'] == ''){
29     $error['email'] = $error_start . "Wypełnij wymagane pole" . $error_end;
30     $valid_form = FALSE;
31 }
32
33 if($error['email'] == '' && !preg_match($regex, $form['email'])){
34     $error['email'] = $error_start . "Wprowadź prawidłowy email" . $error_end;
35     $valid_form = FALSE;
36 }
37
38 if($error['phone'] == '' && !preg_match('^\\+48[0-9]{9}$', $form['phone'])){
39     $error['phone'] = $error_start . "Wprowadź prawidłowy numer telefonu" . $error_end;
40     $valid_form = FALSE;
41 }
42
43 if($valid_form){
44     header("Location: l9p16_success.php");
45 } else {
46     include('l9p16_form.php');
47 }
48
49 } else {
50     foreach($form_fields as $field){
51         $form[$field] = '';
52     }
53     include('l9p16_form.php');
54 }
```

PHP –formularze, podsumowanie

```
1 <!DOCTYPE HTML>
2 <html lang="pl">
3 <head>
4     <meta charset="utf-8"/>
5     <title>Przetwarzanie formularzy -l9p16_success</title>
6     <style>
7         .form1{
8             width:40vw;
9         }
10        .red_label
11        {
12            display:inline-block;
13            background-color:#B60C1B;
14            font-size:14px;
15            color: #fff;
16        }
17    </style>
18 </head>
19 <body>
20     <h3>Twoje dane zostały przesłane</h3>
21
22 </body>
23 </html>
```