

# Wprowadzenie do obsługi formularzy w PHP

---

## Wprowadzenie do obsługi formularzy w PHP

---

Formularze HTML umożliwiają użytkownikom wprowadzanie danych, które następnie mogą być przetwarzane przez PHP. Dane z formularzy można przysyłać na serwer i dalej nimi operować, np. zapisywać w bazie danych, wysyłać e-mailem lub walidować przed wysłaniem.

### Metody przesyłania danych

PHP obsługuje dwie główne metody przesyłania danych formularzowych:

GET – dane są widoczne w pasku adresu przeglądarki.

POST – dane są ukryte i nie pojawiają się w adresie URL.

### Tworzenie podstawowego formularza HTML

Formularz HTML definiujemy przy pomocy znacznika `<form>`, który zawiera pola do wprowadzania danych oraz przycisk do ich przesyłania.

```
<form action="odbior.php" method="POST">
  <label>Imię:</label>
  <input type="text" name="imie">
  <br>
  <label>Wiek:</label>
  <input type="number" name="wiek">
  <br>
  <input type="submit" value="Wyślij">
</form>
```

Opis:

action="odbior.php" – plik PHP, który przetworzy dane.

method="POST" – sposób przesyłania danych.

name="imie" – identyfikator pola używany do odczytu wartości w PHP.

type="submit" – przycisk wysyłający dane do odbior.php.

### Odczyt danych z formularza w PHP

Dane przesyłane z formularza są dostępne w specjalnych tablicach superglobalnych:

#### Odczyt danych metodą GET

```
<?php
if (isset($_GET['imie']) && isset($_GET['wiek'])) {
    echo "Imię: " . $_GET['imie'] . "<br>";
    echo "Wiek: " . $_GET['wiek'];
}
?>
```

Działanie:

Dane są pobierane z adresu URL, np. index.php?imie=Jan&wiek=25.

## Odczyt danych metodą POST

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $imie = $_POST['imie'];
    $wiek = $_POST['wiek'];
    echo "Imię: $imie, Wiek: $wiek";
}
?>
```

Działanie:

Dane są wysyłane w ukryty sposób i nie pojawiają się w adresie URL.

## Walidacja danych w formularzu

Zabezpieczenie danych przed błędami i atakami hakerskimi jest kluczowe. Walidacja może być wykonana po stronie serwera w PHP.

### Sprawdzenie, czy pole nie jest puste

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["imie"])) {
        echo "Pole imię jest wymagane.";
    } else {
        echo "Podane imię: " . htmlspecialchars($_POST["imie"]);
    }
}
?>
```

Działanie:

`empty($_POST["imie"])` – sprawdza, czy pole nie jest puste.

`htmlspecialchars($_POST["imie"])` – zabezpiecza przed wstrzykiwaniem kodu HTML.

## Sprawdzenie, czy wartość jest liczbą

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (!is_numeric($_POST["wiek"])) {
        echo "Wiek musi być liczbą.";
    } else {
        echo "Twój wiek: " . $_POST["wiek"];
    }
}
?>
```

## Walidacja adresu e-mail

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
        echo "Niepoprawny adres e-mail.";
    } else {
        echo "Twój adres e-mail: " . $_POST["email"];
    }
}
?>
```

## Przekierowanie po wysłaniu formularza

Po wysłaniu danych można przekierować użytkownika na inną stronę:

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    header("Location: podziekowanie.php");
    exit();
}
?>
```

## Obsługa checkboxów, radio i select

Formularz może zawierać różne typy pól:

### Checkbox

```
<form method="POST">
    <input type="checkbox" name="zgoda" value="tak"> Akceptuję regulamin<br>
    <input type="submit" value="Wyślij">
</form>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["zgoda"])) {
        echo "Dziękujemy za akceptację regulaminu.";
    } else {
        echo "Musisz zaakceptować regulamin.";
    }
}
?>
```

## Radio button

```
<form method="POST">
    <input type="radio" name="plec" value="Mężczyzna"> Mężczyzna
    <input type="radio" name="plec" value="Kobieta"> Kobieta
    <br>
    <input type="submit" value="Wyślij">
</form>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "Wybrano: " . $_POST["plec"];
}
?>
```

## Select (lista rozwijana)

```
<form method="POST">
    <select name="miasto">
        <option value="Warszawa">Warszawa</option>
        <option value="Kraków">Kraków</option>
        <option value="Gdańsk">Gdańsk</option>
    </select>
    <br>
    <input type="submit" value="Wyślij">
</form>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "Wybrane miasto: " . $_POST["miasto"];
}
?>
```

## Podsumowanie

Formularze umożliwiają przesyłanie danych przez użytkownika.

Dane mogą być wysyłane metodą GET (widoczne w URL) lub POST (ukryte).

Walidacja zapobiega błędom i atakom na serwer.

isset(), empty(), is\_numeric(), filter\_var() pomagają sprawdzić poprawność danych.

Można obsługiwać pola typu checkbox, radio, select.

## Zadania do wykonania

1. Stwórz formularz z polem tekstowym imię i wyświetl jego zawartość w PHP.
2. Dodaj pole wiek i sprawdź, czy użytkownik podał liczbę.
3. Dodaj pole email i sprawdź jego poprawność przy pomocy filter\_var().
4. Dodaj checkbox z pytaniem „Czy akceptujesz regulamin?” i sprawdź, czy został zaznaczony.
5. Utwórz pole radio z wyborem płci i wyświetl wybraną wartość.
6. Stwórz formularz z listą rozwijaną miasto i obsłuż wybór użytkownika.
7. Po wysłaniu formularza przekieruj użytkownika na stronę dziękujemy.php.
8. Zabezpiecz formularz przed pustymi polami używając empty().
9. Stwórz formularz z polem hasła i sprawdź, czy ma co najmniej 8 znaków.
10. Utwórz formularz z polem telefon i sprawdź, czy zawiera 9 cyfr.
11. Stwórz formularz z opcją wielokrotnego wyboru (checkbox) i wyświetl zaznaczone opcje.
12. Stwórz formularz, który wyświetla podsumowanie po wysłaniu.
13. Dodaj zabezpieczenie przed HTML Injection (htmlspecialchars).
14. Wykorzystaj header() do przekierowania użytkownika po poprawnym przesłaniu danych.
15. Stwórz wielostronicowy formularz, gdzie dane z jednej strony przechodzą na kolejną.