

CIASTEczKA I SESIA

1. CIASTEczKA (COOKIES)

- **Czym są ciasteczka (cookies) w kontekście PHP?**

Są to małe pliki tekstowe przechowywane po stronie klienta

Mają za zadanie przechowywać informacje o użytkowniku lub sesji

Wykorzystywane są często do śledzenia zachowań użytkowników na stronach internetowych i zapamiętywaniu danych, które mogą zostać użyte przy kolejnym odwiedzeniu strony

- **Jakie informacje można przechowywać w ciasteczkach?**

Można przechowywać różne informacje, a to są przykłady:

- Dane sesji
- Preferencje użytkownika
- informacje o koszyku podczas zakupów
- dane służące do dostosowywania strony pod użytkownika
- identyfikatory, dane statystyczne, informacje o ruchu na stronie

- **Jak można ustawić ciasteczko w PHP?**

Aby ustawić ciasteczko w PHP, można skorzystać z funkcji `setcookie()`

`setcookie(nazwa, wartość, wygaśnięcie, ścieżka, domena, flaga-bezpieczne, flaga-http-only);`

- nazwa - nazwa ciasteczka.
- wartość - wartość przypisana do ciasteczka.
- wygaśnięcie (opcjonalne) - określa czas wygaśnięcia ciasteczka, wyrażony jako znacznik czasu (timestamp). Jeśli nie jest ustawione, ciasteczko wygaśnie po zamknięciu przeglądarki.
- ścieżka (opcjonalne) - określa ścieżkę na serwerze, dla której ciasteczko jest dostępne. Domyślnie jest to "/", co oznacza, że ciasteczko jest dostępne dla całej witryny.

- domena (opcjonalne) - określa domenę, dla której ciasteczko jest dostępne. Domyślnie jest to bieżąca domena.

- flaga-bezpieczne (opcjonalne) - określa, czy ciasteczko powinno być wysyłane tylko przez bezpieczne połączenie HTTPS. Wartość true oznacza, że ciasteczko jest wysyłane tylko przez HTTPS.

- flaga-http-only (opcjonalne) - określa, czy ciasteczko jest dostępne tylko przez protokół HTTP i nie może być odczytane przez skrypty JavaScript. Wartość true oznacza, że ciasteczko jest dostępne tylko przez HTTP.

- **Jak odczytać wartość ciasteczka w PHP?**

Aby odczytać wartość ciasteczka skorzystać musimy z superglobalnej tablicy `$_COOKIE`

przykład:

```
$wartosc = $_COOKIE["mojeCiasteczko"];  
echo "Wartość ciasteczka: " . $wartosc;
```

- **Jak usunąć ciasteczko w PHP?**

Aby usunąć ciasteczko w PHP, można ustawić jego czas wygaśnięcia na wcześniejszą datę, co spowoduje, że przeglądarka usunie ciasteczko. Można to zrobić przy użyciu funkcji `setcookie()`

przykład:

```
setcookie("mojeCiasteczko", "", time() - 3600, "/");
```

- **Jakie są ograniczenia dotyczące nazw ciasteczek w PHP?**

- Nazwa ciasteczka może zawierać tylko litery (A-Z, a-z), cyfry (0-9) oraz znaki minus (-) i podkreślenie (_).

- Nazwa ciasteczka musi rozpoczynać się od litery lub znaku podkreślenia (_).

- Nazwa ciasteczka jest wrażliwa na wielkość liter, co oznacza, że "ciasteczko" i "CiAsTeCzKo" będą traktowane jako dwie różne nazwy.

- Niektóre znaki specjalne, takie jak spacje, przecinki, kropki, dwukropki, średniki, przecinki, znaki równości i inne znaki specjalne, nie są dozwolone w nazwach ciasteczek.

- Długość nazwy ciasteczka nie powinna przekraczać 4096 bajtów.

- Czy ciasteczka w PHP są bezpieczne? Jakie zagrożenia niosą za sobą?

Ciasteczka w PHP nie są bezpiecznym mechanizmem przechowywania danych, ponieważ są przechowywane po stronie klienta i mogą być odczytane, zmodyfikowane lub podrobione przez użytkowników

Dodatkową mogą powodować kradzież danych, podszywanie się, oraz różne ataki

- Jak mogę sprawdzić, czy ciasteczko istnieje w PHP?

Aby sprawdzić, czy ciasteczko istnieje w PHP, możesz skorzystać z funkcji `isset()`, która również może sprawdzać, czy wartość nie jest równa `null`

przykład:

```
if (isset($_COOKIE["mojeCiasteczko"])) {  
    echo "Ciasteczko o nazwie 'mojeCiasteczko' istnieje.";  
} else {  
    echo "Ciasteczko o nazwie 'mojeCiasteczko' nie istnieje.";  
}
```

PRZYKŁADY:

index.php

```
<form action="index.php" method="post">  
    nick <input type="text" name="nick" id="nick"><input type="submit"  
name="a" value="ustaw"><br>  
    usun <input type="submit" name="b" value="usun"><br><br>  
    <input type="submit" value="odśwież"><br>  
</form>  
<?php  
$cookie_name = "nick";  
if(isset($_POST["a"])){ // przykład ustawienia cookie o nazwie nick  
    $cookie_value = $_POST["nick"];  
    if(empty($cookie_value)) echo "brak danych";  
    else setcookie($cookie_name, $cookie_value, time() + (86400 * 30),  
"/"); // 86400 = 1 day  
}  
if(isset($_POST["b"])){ // przykład usunięcia cookie  
    setcookie($cookie_name, "", time() - 3600, "/");  
    echo "cookie usunięte";  
}
```

```
?>
<br><br>
<?php
if(!isset($_COOKIE[$cookie_name])) { // sprawdzenie czy cookie istnieje
    echo "Cookie-" . $cookie_name . "' jest nie ustawiony";
} else {
    echo "Cookie-" . $cookie_name . "' ustawiony!<br>"; // jeżeli
istnieje to wyświetl co zawiera
    echo "nick to: " . $_COOKIE[$cookie_name];
} ?>
```

2. SESJA ()

- **Czym jest sesja w kontekście PHP?**

Sesja w kontekście PHP to mechanizm przechowywania danych na serwerze, który umożliwia śledzenie i utrzymanie stanu użytkownika między różnymi żadaniami HTTP. Dzięki sesjom możemy przechowywać informacje o użytkowniku, takie jak dane logowania czy preferencje, zapewniając spójność stanu w trakcie interakcji z aplikacją internetową.

- **Jak rozpocząć sesję w PHP?**

Aby rozpocząć sesję w PHP, należy użyć funkcji `session_start()`, która powinna być wywoływana na początku każdego Skryptu PHP

przykład:

```
<?php
session_start();
// Reszta kodu
?>
```

- **Jak przechowywać i odczytać dane w sesji w PHP?**

Aby przechowywać dane w sesji w PHP, możesz korzystać z superglobalnej tablicy `$_SESSION`

przykład:

```
$_SESSION['nazwa'] = 'wartość';
```

Aby odczytać dane z sesji, użyj odwołania do odpowiedniego klucza w tablicy \$_SESSION

przykład:

```
$nazwa = $_SESSION['nazwa'];  
echo $nazwa; // wyświetlenie
```

- **Jak usunąć dane z sesji w PHP?**

Aby usunąć dane w sesji w PHP, można skorzystać z funkcji session_unset() usuwającą wszystkie dane z sesji

przykład:

```
session_unset();
```

można też użyć funkcji unset() do usunięcia danych z sesji, podając odpowiednie klucze w tablicy \$_SESSION

przykład:

```
unset($_SESSION['nazwa']);
```

- **Jak zakończyć sesję w PHP?**

Aby zakończyć sesję w PHP, można skorzystać z funkcji session_destroy(). Ta funkcja usuwa wszystkie dane związane z bieżącą sesją i powoduje zakończenie sesji

przykład:

```
session_destroy();
```

- **Jakie są ograniczenia dotyczące przechowywania danych w sesji PHP?**

- Rozmiar danych: wynosi 128 kB, Jeśli próbujesz przechowywać większe ilości danych, może to prowadzić do przekroczenia tego limitu i spowodować błędy lub utratę danych.

- Czas trwania sesji jest ograniczony: sesja trwa do momentu zamknięcia przeglądarki lub trzeba skonfigurować za pomocą ustawień sesji ile czasu ma istnieć

- Dane z sesji są przechowywane na serwerze więc duże ilości danych wpływają na wydajność serwera

- **Czy sesje w PHP są bezpieczne? Jakie zagrożenia niosą za sobą?**

Dane przechowywane w sesji PHP są przechowywane na serwerze. Jednak sesje mogą być zagrożone przez ataki typu session hijacking lub session fixation, które mogą prowadzić do kradzieży lub manipulacji danymi sesji. Warto zadbać o odpowiednie zabezpieczenia i praktyki programistyczne, takie jak używanie unikalnego identyfikatora sesji i zabezpieczenia przed atakami XSS i CSRF.

- **Czy mogę dostosować ustawienia sesji w PHP, takie jak czas wygaśnięcia i ścieżka sesji?**

W PHP jest możliwość dostosowywania ustawień sesji, takich jak czas wygaśnięcia i ścieżka sesji

- za pomocą funkcji `session_set_cookie_params()`

Pozwala na ustawienie parametrów ciasteczka sesji, takich jak czas wygaśnięcia, ścieżka, domena, bezpieczne połączenie itp. Ta funkcja powinna być wywołana przed rozpoczęciem sesji, idealnie na samym początku skryptu PHP

przykład:

```
session_set_cookie_params(3600, '/');
```

- za pomocą funkcji `session_save_path()`

Pozwala na ustawienie ścieżki, w której będą przechowywane pliki sesji na serwerze. Ta funkcja powinna być wywołana przed rozpoczęciem sesji

przykład:

```
session_save_path('/path/to/session/directory');
```

- **Czy mogę przechowywać tablice i obiekty w sesji PHP?**

PHP możesz przechowywać tablice i obiekty w sesji i robimy tak jak normalnie, czyli za pomocą `$_SESSION`

PRZYKŁADY:

a.php

```
<?php
// Start sesji
session_start();
?>
<form action="a.php" method="post">
    nick <input type="text" name="nick" id="nick"><input type="submit"
name="a" value="ustaw"><br>
</form>
```

```

<form action="b.php" method="post">
    <input type="submit" name="a" value="sprawdź"><br>
</form>
<?php
if(isset($_POST["a"])){ // ustawienie w sesji wartosci
    $_SESSION["nick"] = $_POST["nick"];
    echo "seji zmienna ustawiona.";
}
?>

```

b.php

```

<?php
// Start sesji
session_start();
?>
<?php
if(isset($_SESSION["nick"]))
// pobranie z sesji zmiennej
echo "nick: " . $_SESSION["nick"] ; // i wyswietlenie
else echo "nick nie ustawiony;"
?>

<form action="a.php" method="post">
    <input type="submit" name="b" value="powrót"><br>
</form>

<form action="b.php" method="post">
    <input type="submit" name="usun" value="usun"><br>
</form>
<?php
if(isset($_POST["usun"])){
// usuniecie zmiennej z sesji
unset($_SESSION["nick"])
// usunięcie wszystkich wartosci
session_unset();
// zamknięcie sesji
session_destroy();}
?>

```